An Interop Publication

CONNEXIONS®

The Interoperability Report

March 1993 | *Special Issue: INTEROP 93 Spring Companion* | Volume 7, No. 3

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. 415-941-3399. Fax: 415-949-1779. Toll-free: 1-800-INTEROP. E-mail: `connexions@interop.com`.

ISSN 0894–5926

From the Editor

Welcome to INTEROP 93 Spring and this Special Issue of *ConneXions—The Interoperability Report.* This edition contains articles directly and indirectly related to the conference and tutorial program. *ConneXions* will continue to cover such topics throughout the year, so don't forget to subscribe at the special conference discount rate so you can stay informed.

Our first article is a description of what it takes to build INTEROPnet, the exhibition network to which all vendors are required to connect and demonstrate interoperability with one another. The article is a by Bo Pitsker, Director of INTEROPnet.

INTEROP 93®
8–12 March 1993
Washington, D.C.
Convention Center
SPRING

In the last INTEROP issue (October 1992), we presented an overview of IBM's *Advanced Peer-to-Peer Networking* (APPN). This month Wayne Clark describes an alternative solution, called *Advanced Peer-to-Peer Internetworking* (APPI).

Distributed computing continues to be a hot topic in the industry. Dave Chappell outlines the Open Software Foundation's *Distributed Computing Environment* (DCE).

The Global Internet is a fast growing and exciting place to be. (Just take a look at the Internet Domain Survey on page 44.) New innovative applications are constantly being developed and deployed. *ConneXions* has published many articles in the past on such applications: Archie, Gopher, Netfind, Prospero, WAIS and World Wide Web, to name a few. This month, Carl Malamud describes *Internet Talk Radio* which will start worldwide audio broadcasting in a couple of weeks. Soon you will be able to "tune in" on your workstation and listen to interviews with Internet personalities, or reports from the last IETF meeting.

Our final article is a look at the US LAN/WAN marketplace. The industry is going through rapid changes with many new products—such as bridges, routers, and hubs—and technologies/services—such as FDDI, Frame Relay, SMDS and ATM—being introduced. We asked Nick Lippis, Steve Moore, and John P. Morency of Strategic Networks Consulting, Inc. to summarize the major market trends.

The *Internet System Handbook* was released at the last INTEROP. It, along with many other titles, is available from the *INTEROP Bookshop* in the Washington Convention Center. You'll find a review of this book on page 43.

Insights into the INTEROPnet

by Bo Pitsker, Interop Company

Introduction

The INTEROP INTEROPnet has been well described in previous *ConneXions* articles. However, each implementation of the INTEROPnet is essentially a new exercise in internetworking. Thus it is worthwhile to review the specifics of the network's design and construction, as well as the limitations of the INTEROPnet environment. The INTEROP 92 Spring INTEROPnet was designed with the results of INTEROP 91 Fall in mind; a review of the results will explore the success we had learning the lessons of that network.

The physical context

The Spring event is held in the Washington, D.C. Convention Center. This facility, although relatively new by convention center standards, was not designed for easy installation of communications cabling. For the spring show, we were using Halls A and B only for exhibits; that will change this year with the addition of Hall D. Hall A has a 40-ft. high curved ceiling and is column free. Hall B, the larger hall, has a 30-ft. high ceiling with rectangular columns on 90-ft. centers. Our challenge was to design a network cable plant which could be installed in multiple segments, including an interconnect between the two halls, and including connections to the NOC (*Network Operation Center*) on the opposite side of the hall from the backbone location. Approximately 280 network drops had been requested by exhibitors. Terminal clusters were to be located in the upstairs and downstairs lobby areas, and via a remote link to the Hyatt Regency Capitol Hill Hotel, several miles away.

General design goals

It is worth making explicit the general design goals for the INTEROPnet. First, the network is intended to be the vehicle for each exhibitor to demonstrate some form of interoperability outside the exhibitor's booth. When INTEROP was a TCP/IP interoperability conference, this was relatively straightforward. Now that INTEROP encompasses LAN/WAN technologies across multiple media, protocols, platforms, and standards, complete interoperability across the INTEROPnet becomes more difficult to achieve.

Second, the INTEROPnet is, to some extent, the victim of its own success. Many exhibitors expect that the network will always be up; that is, the INTEROPnet has evolved into a production network, as distinct from a demonstration facility. Indeed Interop has encouraged that view by offering *Solution Showcase*™ Demonstrations, where vendors build a separate demonstration which may or may not involve use of the INTEROPnet. Frequently technologies first shown in a Solution Showcase are migrated to the INTEROPnet. Recent examples include FDDI, OSPF, and SNMP. Clearly the requirement for high availability constrains incorporation of new technologies and limits the possible service offerings.

Third, the INTEROPnet has to be constructible within the context of industry support and in a trade show environment. Over 50 vendors donated equipment or personnel to the effort; the inventory alone ran to more than 1200 items. The volunteer participation exceeded 200, with many contributing not days, but weeks to the effort. A network requiring even more than this level of effort isn't buildable by Interop or anyone else.

The trade show environment is worthy of discussion, as comments are sometimes made that the effort required to build the INTEROPnet doesn't seem commensurate with the final results.

What is frequently overlooked is that we had less than a week in the Convention Center to construct the cable plant, and 24 hours to install it immediately prior to the opening of the show, and 72 hours to bring everything on-line, using individuals who may not have familiarity with the equipment or each other. This is *not* your normal enterprise-wide project.

Specific design goals

Our design of the Spring network began last Fall with discussions of what worked and what didn't for that INTEROPnet. Chief among the concerns was the desire to improve overall reliability of the network. Two problems had manifested themselves during operation of the 91 Fall INTEROPnet. The first was multiple failures of Internet connectivity via BARRnet. There wasn't any redundancy, as there was only a single line to the Internet, so lost T-1 carriers, router crashes, etc. all contributed to a perception of significant downtime. This led to our first specific design goal: redundant access to the Internet. We accomplished this with 3 T-1 circuits; 2 to Alternet at Falls Church, VA, and 1 to Alternet at College Park, MD. We even had diversity routing, as the Fall Church lines went via C&P, while the College Park line went via Metropolitan Fiber Systems.

The second reliability problem concerned router failures severe enough to require rebooting or even power cycling. The causes for router failure varied, but included routers running multiple protocol stacks, not all of which were as mature in their implementation as were the IP stack, and the CPU/memory demand placed on routers by the Fall configuration, which had a router for every rib. In practice, the latter design element created lightly loaded router traffic but very large routing tables, due to the adjacency requirements.

Two steps were taken to alleviate this situation. First, the fan-out of router ports was increased so that a single router supported six ribs. This increased the loading but decreased the size of the routing table information to be propagated. Second, a dual router configuration was implemented. Each router location contained two routers, one running IP only and the other running both IP and other protocol stacks. Both routers would "see" the same ribs, but the IP-only router would be the preferred route for IP while the other router was the only path for non-IP protocols. Should the IP router fail, the non-IP router would acquire the IP routes via OSPF. Note that this design still required a manual reconfiguration in the event that the non-IP router died.

Protocols and topology

Another design goal was to expand the protocols supported. As the show continues to grow, the requests for additional protocol stacks continues to increase. OSI was a natural, as Washington D.C. can be considered the capitol of OSI activity in the United States. Beyond OSI, IPX (NetWare) and AppleTalk were selected as protocols where significant support was available to meet the demand.

Finally, Token Ring and FDDI were utilized. This time, rather than give every exhibitor a "hot" Token Ring connection automatically, we furnished it to only those who asked for it. Because our experience has been that Token Ring is not utilized at INTEROP to the same degree that it is installed in the industry, we did not provide a separate ring on every rib. Instead Hall A became a single ring, and Hall B became another ring. The connection between them was via a router. This decision to route, not bridge, the two rings, was to have important consequences later.

continued on next page

Insights into the INTEROPnet *(continued)*

FDDI was used in essentially the same configuration as in INTEROP 91 Fall; i.e., the dual ring consisted of concentrators or single attached router ports only. For the first time the ring was extended to the NOC and attached to various diagnostic equipment.

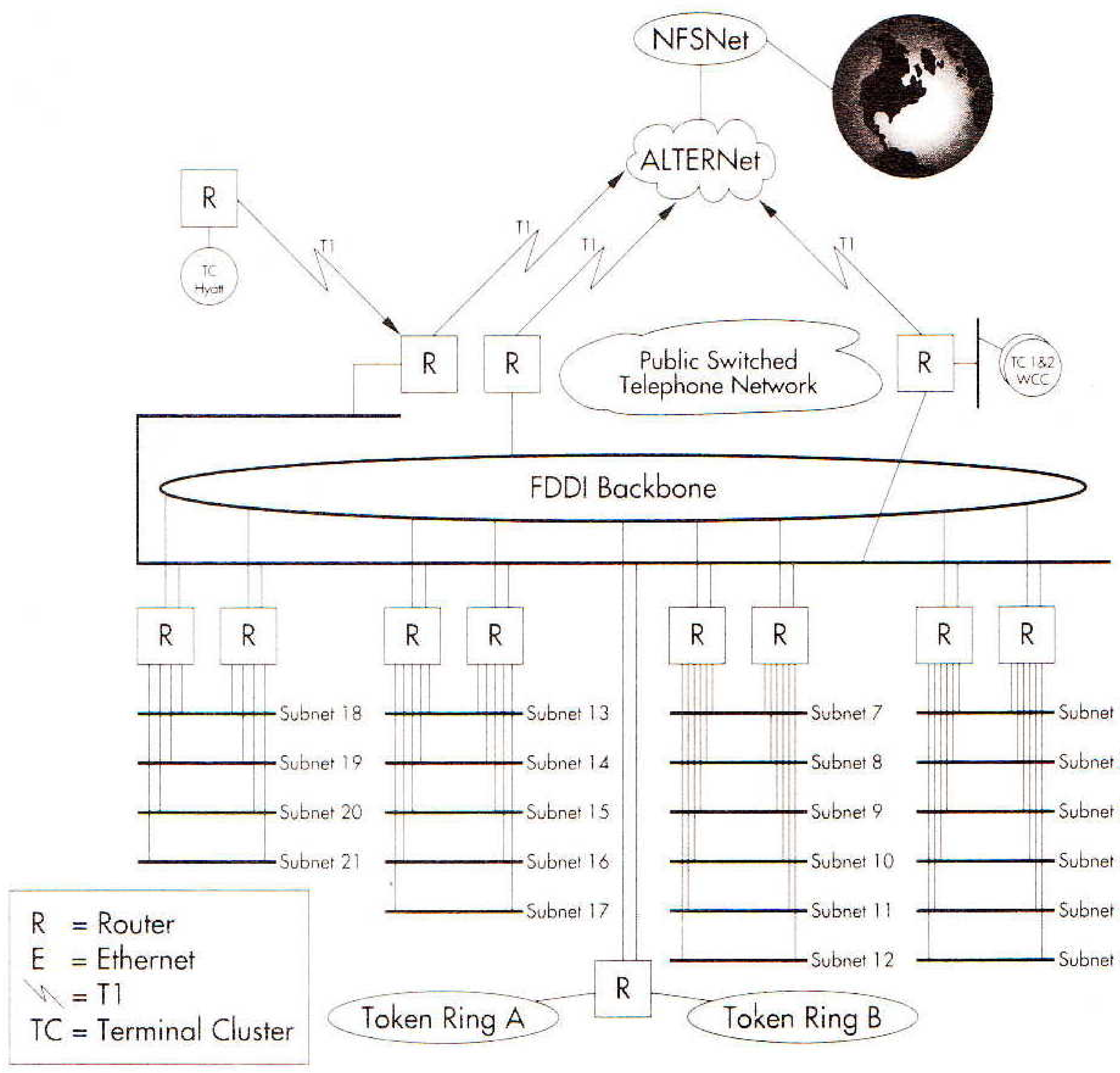

Figure 1: INTEROP 92 Spring INTEROPnet Topology

Network management

Previous INTEROPnets depended almost exclusively on standard low-level (i.e., topology-ignorant) tools such as *ping, traceroute,* and *telnet.* While these continued to play a central role, a concerted effort was made to do some side-by-side comparisons with network management tools, especially those using SNMP. To that end a room in the NOC was designated as the "Network Spy Atrium," or NSA, and network management and protocol analysis workstation vendors were solicited for contributions. We did receive substantial contributions, and the results proved interesting.

In addition, so-called "spy wires" have always been used, bringing a link up from each rib so that protocol analyzers could be used, traffic pushed onto a particular rib/router port, or a router could be bypassed entirely, if necessary, etc. However, with the NOC across the hall from the backbone, the distances proved too long for 10Base-T, and the necessity of using separate repeaters for each rib was inelegant. Two solutions were tried: the first involved using an intelligent switch box with a single repeater to select a line for analysis by remote control. The second was to use remote protocol analysis units located in each pedestal. Both solutions were implemented. A third partial solution was to configure a centrally-located router with a port for every rib, and compare what it saw vs. the default router. This router was referred to fondly as "the spy router." The spy router provided insight about whether another router's interface was wedged or an actual problem existed on a rib. However, protocol analyzers were still useful; a spy router by its nature can't provide the same glitch detection.

Implementation

The life cycle of the INTEROPnet consists of design, pre-wire, hot-staging, installation, operation, and teardown. This is accomplished within a 5–7 month timeline, which overlaps with activities supporting previous and subsequent conferences. The order of magnitude can be described in part by some of the statistics: 1,100 pieces of equipment installed; 60 miles of UTP, STP, and FDDI; over 1,700 exhibitor hosts were detected by the end of the show.

Equipment began to arrive at Interop's Sterling, Virginia warehouse at the end of March. By the third week of April most of it was mounted in 26 EIA-standard racks, and configuration had begun, both by on-site Interop and volunteer engineers, and remotely, via an Alternet T-1 drop.

Setup and testing was interrupted by pre-wire at the Washington D.C. Convention Center, euphemistically billed as "92 Spring Network Design Fiesta," and accompanied by numerous tee shirt competitions, surely won by InterCon's prodigious output and sometimes risque content. Pre-wire ran from Wednesday, April 15, through Monday, April 20. During this event, all of the ribs and most of the backbone was constructed and "flown," that is, hung from the ceiling, in order to verify the cable plant design and make adjustments. Obstacles overcome included having the Center lights turned off after 6 pm; work continued under massive generator-driven halogen flood lamps.

The following week saw the return of the newly-constructed cable plant, together with the NOC team and vendor engineers, to the warehouse for 10 days of hot staging. Routers were configured and tested, first IP, then other protocols; mysterious terminal behaviors were deciphered, differing pinouts for dozens of serial ports and devices were slowly understood, documented, and cabled; rib-by-rib testing exercised the intervening network equipment. Some equipment failed and was returned to the vendors; others required ROM replacements, new software loads, or non-standard configurations. Bugs that could not be corrected were documented so that workarounds could be constructed, or features dropped. Especially helpful during this testing were tools designed to stress networks or validate correct behavior. Examples included PCs running packet generator software, specialized test generators such as Wandel & Goltermann's DA-30, Network General's Sniffer, and NIST's OSI test suite, built on Sun platforms.

Actual installation of the INTEROPnet began Saturday, May 16, at 8 am. The goal was to have the cable plant in place before 8 am Sunday, as that was the beginning of freight move-in. The move-in requires that all network wiring be suspended above 14 feet; otherwise the semis have a good change of pulling it down. We tried a new method of doing this, by putting RJ-45 connectors at 14 feet, then attaching "tails" later. The installation was completed by midnight. Connection and testing of the pedestals continued through Sunday, and exhibitor hookup began Monday, completing Tuesday night. Operation commenced Tuesday night, and continued up to the closing time of 5 pm Friday.

Observations

Overall, the redesign of the INTEROPnet was successful. While there were instances of router port lockups and one router crash, the overall network stayed up throughout the show. OSPF proved to be capable of proving reasonably quick failover and subsequent route redistribution.

The FDDI backbone stayed up throughout the show, and was the preferred route for traffic, with the Ethernet serving as backup only. Consequently, the majority of data passed through the FDDI ring, which constitutes a milestone for INTEROPnet. The design standardized on SMT 6.2; it will be interesting to observe how long implementations of SMT 7.x will take to achieve equivalent stability.

continued on next page

Insights into the INTEROPnet *(continued)*

The decision to route between token rings overlooked several exhibitors who needed to pass SNA traffic between them and the SNA-TCP/IP Solution Showcase Demonstration. We discovered the situation too late to engineer a bridged solution, and ended up cabling around the problem. A likely future solution will consist of bridges filtering on SAP values for SNA, thus minimizing the risk of "leaking" routeable packets across the bridge.

The network management tools proved to be a mixed lot: those that used SNMP sometimes had a different view of the network state than did those that didn't. The learning curve on some of the units was daunting enough to preclude a casual, trial-and-error means of becoming familiar with the capabilities off the system. For most NOC team members this meant that they left the units in the hands of the manufacturers' reps who staffed them.

However, when set up and maintained by the reps, the network management did furnish useful statistics about the performance of the network, and many faults were detected. The ability to capture and report back detailed configuration information varied to an extreme—apparently because either the resident database wasn't sufficiently "rich" in product information, or because the data could only be acquired via private MIBs.

Summary

The successful design and implementation of a network of this size and complexity has important implications. A more typical scenario for the provision of an enterprise-wide network of this magnitude would have it designed by one group, installed by another, and the entire process would be stretched out over years. The costs, both direct and indirect, would be much higher, the results less closely coupled to user requirements, and the likelihood of error higher.

The success of INTEROPnet is attributable to (1) first-class technical help; (2) an artful blend of new and stable technology; and (3) very fast deployment. The first point cannot be stressed enough. The NOC team, although numbering only a dozen, consisted of individuals who were not only experts in various aspects of networking, but who also had significant project management and consulting experience. As has been well-documented with programmer productivity, the productivity of a small team of highly-trained, highly motivated professionals vastly exceeds that of a much larger organization.

The mix of stable and new technologies is an interesting one. The approach adopted by the NOC team was to try technologies which had some track record already, which had some representative or advocate on the NOC team itself, and for which alternatives were in place or could be quickly improvised in the event of failure. This strategy has proven its worth.

From start to finish INTEROPnet required only 5 months. While this time frame was relentless in its pace, the great advantage was that group memory was preserved, alternatives were tested quickly and adopted or discarded, and the network wasn't obsolete by the time it was turned on. For a corporation or institution contemplating migration to a modern network, the conclusion is as obvious as it is infrequently heeded: put together a small team of experts, give them highest level goals and objectives, then tell them that they have absolute authority to get the job done. It will get done, faster and better than the traditional "white paper" methods. After all, you saw it done at INTEROP!

Postscript

This article was prepared after the 92 Spring event. While the overall design goals for INTEROP 93 Spring are similar, there are several notable changes. First is the addition of the downstairs exhibit area, Hall D. This space has columns every 30-ft., with ceiling clearances of only 14-ft., and no overhead structural steel. Accordingly, we have fabricated steel frames to gird the columns, providing us with support for our ribs.

The interconnect between the upstairs and downstairs halls will be provided by a custom composite fiber cable running from the NOC to a central pedestal in Hall D. This arrangement is similar to that used in 92 Fall at Moscone Center to connect the North and South areas together. The fibers will carry FDDI, Ethernet via FOIRL, Token Ring, Frame Relay and SMDS traffic.

Another architectural change is the collapse of several ribs into a single subnet (see Figure 2). This was implemented to reduce the number of router ports required. We found, in the 92 Fall show, that this change did not result in long-term saturation of any segments. It did force us to use 10-bit subnet masks, much to the consternation of some exhibitors.

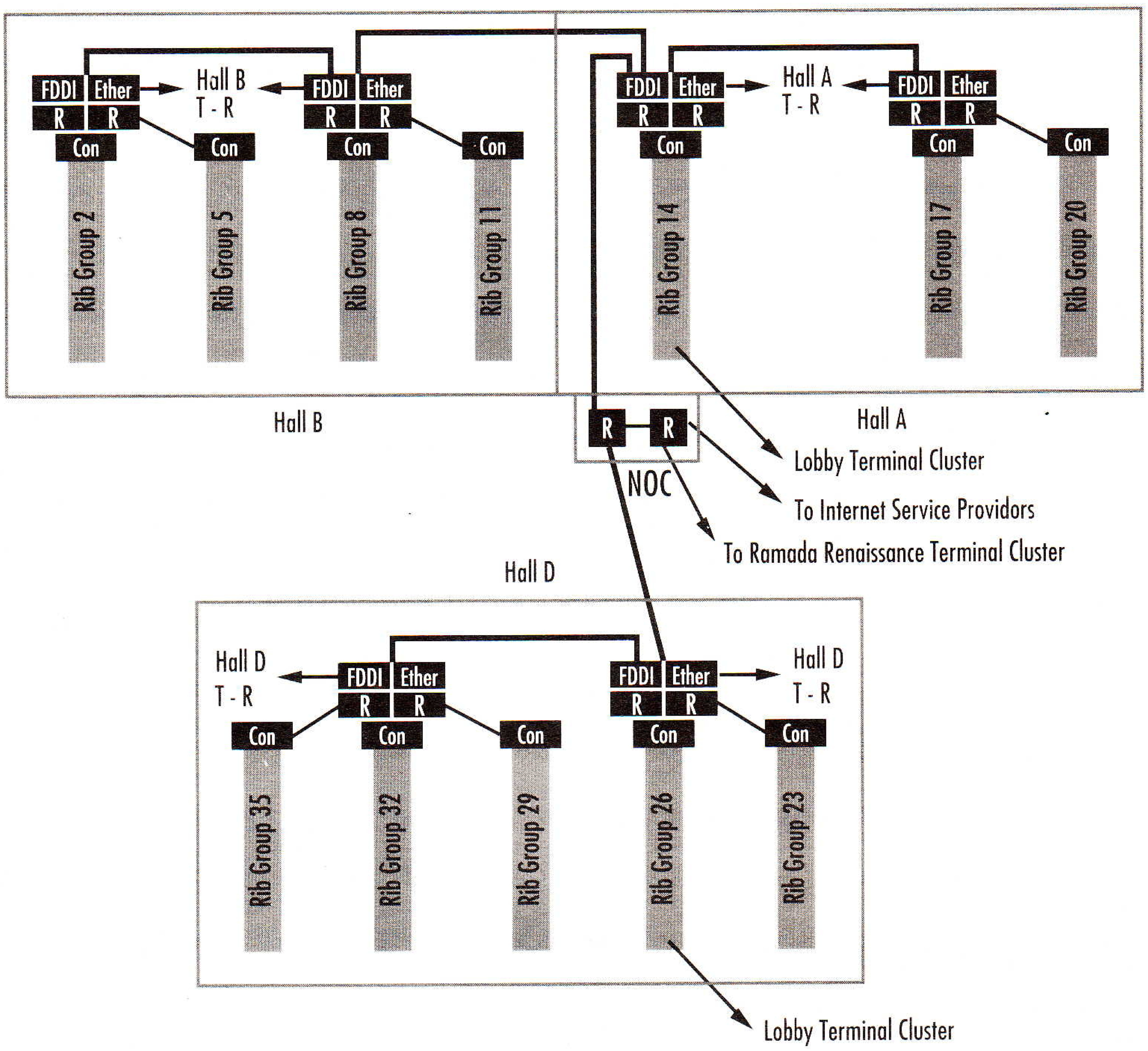

Figure 2: INTEROPnet topology for INTEROP 93 Spring, showing backbones, rib groups and external connectivity.

Finally, we are offering additional protocols, notably DECnet and SNA; the latter being transported via Token Ring. Each hall's rings will be bridged via source-routing to permit the flow of SNA frames.

As always, the INTEROPnet continues to educate attendees on the benefits of internetworking, provide a means for exhibitors to demonstrate their products and services, and demonstrate interoperability on a grand scale.

continued on next page

Insights into the INTEROPnet *(continued)*

References

[1] Jacobsen, O., "INTEROP 88 Conference Report," *ConneXions,* Volume 2, No. 11, November 1988.

[2] Almquist, P., "The INTEROP 88 Network—behind the scenes," *ConneXions,* Volume 3, No 2, February 1989.

[3] Dern, D., "Highlights from INTEROP 89," *ConneXions,* Volume 3, No. 11, November 1989.

[4] Knowles, S., "The INTEROP 89 Network: From one of its builders," *ConneXions,* Volume 4, No. 2, February 1990.

[5] Dern, D., "INTEROP 90 Wrap-Up: A View from the Floor," *ConneXions,* Volume 4, No. 11, November 1991.

[6] Knowles, S., "Building the INTEROP 90 Show Network," *ConneXions,* Volume 5, No. 9, September 1991.

[7] Dern, D., "'Dear Cliff': A Report From INTEROP 91 Fall," *ConneXions,* Volume 5, No. 12, December 1991.

[8] Hares, S., "Lessons Learned for OSI at INTEROP 91 Fall," *ConneXions,* Volume 6, No. 3, March 1992.

[9] Chapman, B., "Building & Managing the INTEROP 91 Fall Shownet," *ConneXions,* Volume 6, No. 6, June 1992.

[10] Dern, D., "INTEROP 92 Spring Proves a Capital Idea," *ConneXions,* Volume 6, No. 7, July 1992

BO PITSKER holds a B.A. in Anthropology and History, with graduate work in Linguistics, Social Anthropology, and Computer Science. As Director of INTEROPnet for Interop Company, he is responsible for the design, installation, and operation of the network. Mr. Pitsker was Manager, Network Installation and Support, for JWP Information Systems, and was responsible for pre-sales engineering, post-sales installation and support for Novell, 3Com, Synoptics, and other LAN and WAN products. Mr. Pitsker was employed at Lawrence Livermore National Laboratory for 8 1/2 years, with varied responsibilities including scientific software development on a variety of hardware platforms, editing of a microcomputer journal, and pioneering desktop publishing. Bo has been a member of the IEEE Computer Society and IEEE Professional Communications Society, and has published numerous articles in the trade literature. E-mail: `bo@interop.com`.

INTEROP Volunteers

Interop Company offers several opportunities for students and others to participate in the conference and exhibition program. You can become an *INTEROPnet Volunteer* (see article above) and participate in the design, installation and operation of INTEROPnet. Contact Bo Pitsker (`bo@interop.com`) for more information.

The Programs department of Interop Company has a *Conference Assessment Team* (CAT) program especially designed for students in the field of computer science. As a CAT member you will be asked to monitor and report on a number of conference sessions. Contact Ole Jacobsen (`ole@interop.com`) for more information.

Interop Company
480 San Antonio Road, Suite 100
Mountain View, CA 94040–1219
Phone: 415-941-3399 or 1-800-INTEROP • Fax: 415-949-1779

Accommodating SNA Peer-to-Peer Networking in a Multiprotocol Environment

by Wayne Clark, Cisco Systems

Background

Even though IBM's *Advanced Peer-to-Peer Networking* (APPN) has existed since 1985, it has only been deployed onto a limited number of platforms and achieved success in small- and medium-sized AS/400 networks. While the migration away from host-centric applications toward peer-to-peer applications is an admirable goal, numerous doubts have been raised about the viability of APPN as the backbone protocol.

This article presents the alternative being pursued by the *APPI Forum* in order to accommodate APPN in a multivendor, multiprotocol network. This alternative, called *Advanced Peer-to-Peer Internetworking* (APPI™), is open to the industry and interoperates with APPN End Nodes.

Introduction

IBM's hierarchical SNA has long been the networking architecture of choice for the commercial computing environment. When *Systems Network Architecture* (SNA) was first introduced in the early 1970s, it had vastly greater capabilities than other networking alternatives of its day. However, SNA hasn't really kept pace with the price/ performance achievements that have been afforded other protocols, especially TCP/IP.

Over the past several years, SNA has been slowly evolving to meet the demands of a distributed computing environment. The pace of that evolution has increased significantly over the past year with the publication of IBM's "Networking Blueprint." The SNA cornerstone of this blueprint is IBM's Advanced Peer-to-Peer Networking (APPN).

APPN is an IBM proprietary protocol with limited backward compatibility with older SNA networks. The proprietary nature of APPN has cast much doubt on the viability of the APPN strategy in today's multivendor, multiprotocol world.

In order to entice more vendors to provide APPN services, IBM has offered to license the APPN Network Node source code to the networking industry beginning in the first quarter of 1993. While this move will definitely make APPN services more readily available in the industry, it does not solve the real problem of multiprotocol support within today's corporate networks.

APPI Forum

The APPI Forum was organized by Cisco Systems and the foundation meeting for the Forum was held during INTEROP week in October 1992. The Forum is a consortium of companies whose charter is to provide the industry with an open alternative for supporting APPN nodes within a multiprotocol network.

An understanding of the services provided in a native APPN network is critical when contrasting APPN with the open APPI alternative. Therefore, this article presents the open alternative to APPN by first providing a general background on native APPN, then discussing the overall architecture of the open solution that the APPI Forum is pursuing, followed by a description of how this open solution fits into today's multiprotocol network.

Overview of APPN

APPN is a peer-based form of SNA used to connect different IBM machines together in an arbitrary topology. APPN was originally designed to interconnect relatively small systems together in a manner that permitted autonomous control and ease of use [1].

continued on next page

APPI *(continued)*

This autonomy is achieved in APPN with the dynamic definition of network resources and the ease of use is addressed with a protocol that permits the automatic discovery of these network resources. (SNA network resources in the context of APPN are either *Logical Unit*(LU) names or *Control Point*(CP) names.)

An excellent overview of APPN was presented in the October 1992 issue of *ConneXions*[3]. For a historical perspective on SNA and APPN, please refer to the March 1992 issue of *ConneXions* [4].

The dynamic aspects of APPN result in a number of different network broadcasts between APPN nodes. For performance reasons, the APPN network is constructed in a two-tiered hierarchy. APPN *Network Nodes* (NN) are inner nodes within the APPN network that contain knowledge about the topology of the network and can perform intermediate session routing. These NNs are analogous to intermediate routers in a multiprotocol network or *Intermediate Systems* (IS) in an OSI network. APPN *End Nodes* (EN) are "leaves" attached to the Network Nodes and do not contain any knowledge about the topology of the network. A special kind of End Node known as a *Low Entry Networking* (LEN) node also connects to Network Nodes but requires static definition of network resources.

A sample APPN network is shown in Figure 1 below. Even though this diagram is drawn with point-to-point links between adjacent APPN nodes, the actual connections may be across a broadcast medium such as a Token Ring LAN.

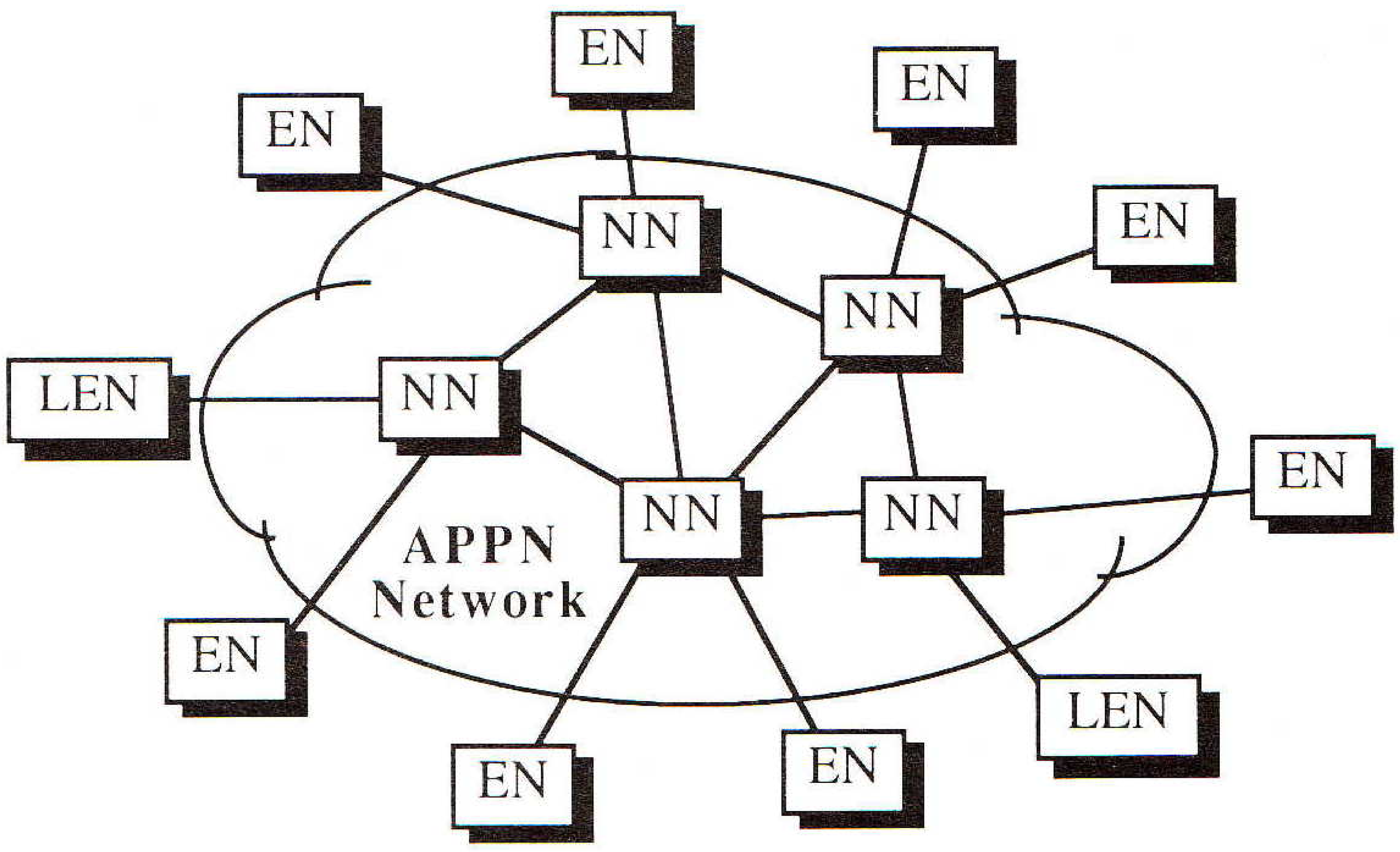

Figure 1: A Sample APPN Network

APPN Node Types

APPN End Nodes (EN) are SNA end systems that can be the source and/or destination of SNA data flow but do not provide any routing services. End Nodes are always full SNA nodes that provide services to end-user applications hosted on the node. End Nodes rely exclusively on their directly-connected Network Nodes for APPN services. The APPN services that a Network Node provides to its End Nodes are:

1. Dynamic registration of SNA resources,
2. Dynamic discovery of SNA resources, and
3. Automatic selection of route.

End Nodes participate with their Network Node in order to implement 1 and 2 while item 3 is implemented entirely by the network of NNs and communicated as needed to the End Node.

APPN Low Entry Network Nodes (LENs) are SNA end systems similar to End Nodes but LEN nodes cannot rely on Network Nodes for APPN services. While LEN nodes are still directly attached to Network Nodes, the LEN node must contain static definitions of all SNA resources that it needs to access. All of these resources appear to be in the Network Node, and the NN assumes responsibility for getting the data to the correct node. Likewise, all of the resources in the LEN node that are accessible from the APPN network must be statically configured in the adjacent Network Node.

Finally, APPN Network Nodes are SNA intermediate systems that perform route selection and provide directory services to other APPN nodes. Network Nodes can be either full SNA nodes providing services to end-user applications hosted at the node (e.g., an AS/400) or can merely be intermediate routing nodes for an APPN network (e.g., a 3174).

The APPN services that a Network Node provides to other Network Nodes are:

- Dynamic registration of SNA resources,
- Dynamic discovery of SNA resources,
- Automatic selection of route, and
- Dynamic updates to changes in topology.

The APPN *Border Node* is a special implementation of the APPN Network Node that is used to connect multiple APPN networks. At the present time, Border Node is only implemented on the AS/400.

The APPN *Composite Network Node* will be the manifestation of the APPN Network Node in VTAM and NCP sometime in 1993. The combined image of VTAM and NCP becomes a single (i.e., composite) Network Node to rest of an APPN network. A Composite Network Node can contain a centralized directory database known as the *Central Directory Service* to reduce the number of network broadcasts for directory searches.

Disposition of the APPN Architecture

As of the writing of this article, APPN has been implemented on a variety of different IBM platforms. Implementations on some systems can be configured to be any one of the three different types of APPN nodes, while other implementations are fixed to be only one particular type of node. Outside IBM, there are a number of implementations of LEN node, but no implementations (yet) of either the APPN End Node or Network Node. Systems Strategies will have the first non-IBM offering of an APPN End Node in early 1993.

IBM has published the APPN End Node architecture in the latest revision of the *Type 2.1 Node Reference* (SC30-3422-2). IBM has stated that it will not publish the APPN Network Node architecture in this form but will provide Network Nodes specifications for a yet-unannounced license fee. IBM will also license the APPN NN source code to outside communications vendors beginning the first calendar quarter of 1993.

The table on the following page summarizes the world of APPN as of the writing of this article. The table includes not only implementations of the APPN architecture from IBM and non-IBM sources but also includes information about future releases of products. A √ means that the implementation is already available.

continued on next page

APPI *(continued)*

A *no* means that there has been a public statement indicating there will be no implementation, a *yes* means that there has been a public statement indicating there will be an implementation, a *?* indicates some doubt on the part of the author (usually the result of lack of definitive movement toward the implementation), a *SOD* (Statement of Direction) indicates a formal statement of the intent to implement, and *n/a* means not applicable (not in the line of business for the particular vendor). A *date* indicates the year the implementation is planned to be available.

	Implementation	LEN	EN	NN
IBM	AS/400	√	√	√
	NS/2	√	√	√
	OS/2 EE CommMgr	√	no	no
	3174 w/ Config Sppt C	no	why not?	√
	VTAM/NCP	√	1993	1993
	6611	n/a	n/a	1993
	RS/6000	√	1993	1993
	OEM	no	yes?	1993
	NS/DOS	√	no	no
Non-IBM	System Strategies, Inc. (OEM)	√	1993	n/a
	Data Connections Ltd. (OEM)	√	1993	?
	Novell - Netware for SAA	√	SOD	1993?
	Eicon - SNA LAN Gateway	√	n/a	n/a
	DCA - Select Comm Server	√	n/a	n/a
	3Com Corporation	no	no	1993
	N.E.T.	no	no	?
	Apple Computer	√	SOD	?

Table 1: APPN Implementations

Overview of APPI

The APPI Forum is advocating an open architecture that accommodates APPN nodes but within the context of a multivendor, multiprotocol network. This is in lieu of the proprietary networking scheme advocated by IBM with the licensing of APPN Network Node source code.

From the preceding discussion, it should be noted that the intermediate APPN network comprised of Network Nodes provides two, very general services to the nodes at the edges of the network (i.e., the End Nodes and LEN Nodes):

- Intermediate session routing through knowledge gained about network topology, and
- Distributed directory services.

All of today's network architectures provide similar network services and have the added benefit of accommodating multiple protocols rather than just SNA.

The APPI Forum is calling the open architecture that accommodates APPN, *Advanced Peer-to-Peer Internetworking* or APPI™. The essence of APPI is that while the edges of an APPN network in the form of APPN End Nodes and LEN Nodes are accommodated in native APPN fashion, the core of the network is multiprotocol in nature. The primary focus of APPI is to provide APPN Network Node services to End Nodes while retaining the flexibility of today's multiprotocol network.

The architecture of an APPI-based network looks identical to an APPN network on the fringes but takes on the characteristics of a true dynamic, multiprotocol network on the inside. The isolation between the APPN network on the fringes and the multiprotocol network in the middle is established through a special facility on a multiprotocol router that provides a barrier between the two networks. A router that provides this isolation is termed an *Open Network Node* or ONN.

An ONN communicates with APPN End Nodes and Low Entry Networking Nodes using LU 6.2 and NT 2.1 protocols, but communicates with the multiprotocol network (including other ONNs) using TCP/IP protocols. By speaking SNA out one side of the ONN and accommodating the EN-to-NN protocol, APPI does not require any changes to the APPN End Nodes on the perimeter of the multiprotocol network.

While an ONN does have a resident SNA protocol stack, it does not need all of the complexity of a full APPN Network Node. The LEN level of functionality in NT 2.1 along with LU 6.2 is an adequate starting place for developing an ONN. The extensions to the LEN architecture that are needed in order to communicate with ENs and LENs are minor when compared to the complexity of APPN NN implementation. These extensions to LEN will be published by the APPI Forum.

The picture of an APPI network, shown in Figure 2, is contrasted with the APPN network, shown previously in Figure 1. In an APPI network, all APPN nodes are either End Nodes or LEN Nodes. The functionality afforded the APPN network by the APPN Network Nodes is supplanted with the ONNs and a number of interconnected multiprotocol routers.

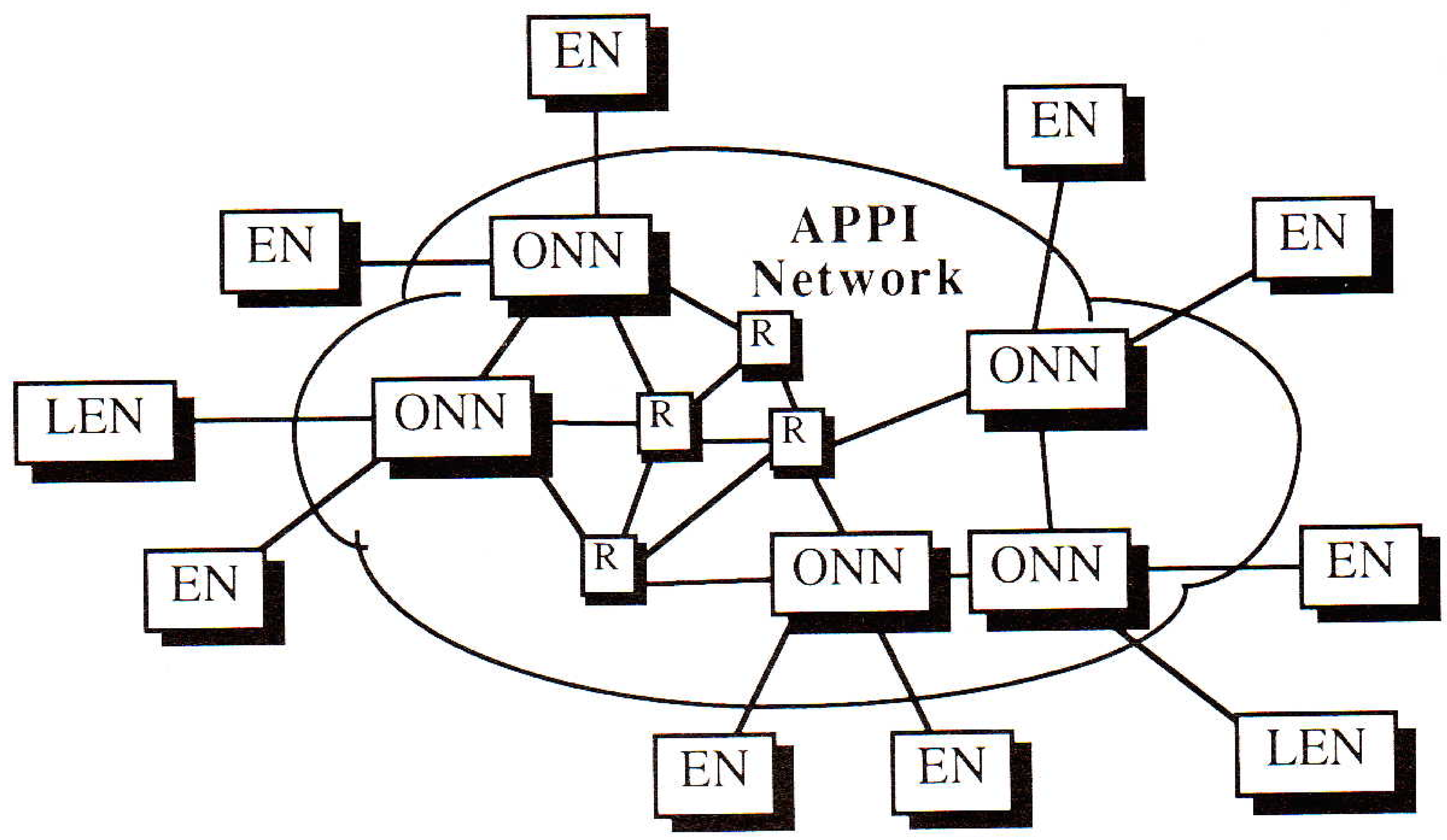

Figure 2: A Sample APPI Network

It should be noted that even though Figure 2 appears to imply that the ONNs are somehow different than the intermediate routers in the APPI network, in reality, an ONN is nothing more than a router or workstation running the special software that provides APPN-like services to the APPN End Nodes.

APPI Routing Protocol

When two different network routing protocols are used in a single network, information obtained in one routing protocol must be distributed into the domain of the other. For example, when *Open Shortest Path First* (OSPF) and the *Interior Gateway Routing Protocol* (IGRP) are both used in a single network, routing information discovered through IGRP must be distributed into OSPF's update messages.

continued on next page

APPI *(continued)*

The topic of redistribution of routing protocol information is not an issue with APPI since the routing protocol used in an APPI network is simply the routing protocol of choice within the multiprotocol network. The proprietary APPN routing protocol no longer exists. (In an APPN network, the cumulative set of APPN Network Nodes when taken as a unit embody the APPN routing protocol. Since APPI entirely eliminates the need for APPN Network Nodes, there are no special considerations that must be given to adapt the routing protocol of APPN to the routing protocol of the multiprotocol network.)

APPI Directory Services

One of the only advantages that APPN networking has over its modern multiprotocol counterparts is a distributed directory service that is tightly integrated with the routing protocol. This level of integration does not yet exist in either TCP/IP or OSI even though both of these architectures have both routing protocols and distributed directory services. TCP/IP routing protocols—OSPF, RIP, IGRP, and Integrated IS–IS—are separate from the *Domain Name System* (DNS) while OSI's routing protocol—IS–IS is separate from X.500.

This lack of integration of a routing protocol with its distributed directory service is addressed in APPI's *Distributed Directory Service* (DDS) protocol. It is more robust than DNS yet not as complex to implement as X.500.

Since the APPI Forum is committed to open standards, it is our desire to use standard services wherever possible. APPI Forum members intend to implement the APPI-specific DDS while concurrently working with the *Internet Engineering Task Force* (IETF) to enhance DNS.

APPI Data Transport

In an APPN network, the path taken between two End Nodes is composed of a series of hops through intermediate Network Nodes. In order to transport data end-to-end between a pair of End Nodes, the APPN network must manage a series of concatenated SNA sessions—one session between each pair of APPN nodes in the route. The protocol overhead required to manage these hop-by-hop sessions effectively limits the throughput of native APPN data transport services.

APPI data transport services, on the other hand, builds upon existing data link transports in order to transfer data between a pair of End Nodes. This not only eliminates the throughput problem associated with managing hop-by-hop sessions but permits SNA traffic to be carried over a multiprotocol network. By accommodating End Node traffic in this manner, SNA sessions are provided the same level of dynamic routing that TCP/IP sessions are given.

The main difference between APPI's data transport services and the data transport services provided by all of today's multiprotocol tunneling features is the level of granularity of the SNA traffic. Data link tunneling features (such as IBM's Data Link Switching and Cisco's serial tunneling and remote source-route bridging) merely replace a point-to-point connection between two SNA end systems with a multiprotocol network, therefore routing data at the SNA *Physical Unit* level.

APPI's data transport services examines each SNA frame and makes routing decisions based upon the SNA session. SNA sessions are established by Logical Units so APPI can be viewed as providing *Logical Unit* routing. This level of sensitivity affords APPI the same granularity as APPN without APPN's shortcomings (such as the inability to route around intermediate link failures).

Management of an APPI network

Since APPI is an open architecture based upon TCP/IP networking services, an APPI network will be managed with the *Simple Network Management Protocol* (SNMP). The *Management Information Base* (MIB) that is being defined for an Open Network Node contains objects for the SNA-side of the ONN router as well as objects that are specific to the APPI functions such as the Distributed Directory Service. As with all portions of the APPI specification, the APPI MIB will be published and made generally available.

APPI versus APPN-over-TCP/IP

APPN running over TCP/IP was demonstrated at INTEROP 92 Fall and was offered as a solution that is essentially the same as APPI. This section discusses APPN-over-TCP/IP in detail and contrasts it with APPI.

What has been done in the APPN-over-TCP/IP offering is to replace SNA's traditional data link controls such as SDLC, LLC2, and QLLC with a reliable TCP/IP network connection. Rather than interfacing the SNA stack directly to one of its usual data links, the SNA stack was placed on top of a *sockets* interface which, in turn, uses TCP/IP as its transport. As a consequence, there are two different protocol stacks on each APPN node in the network that uses TCP/IP as its transport.

More ominous than dual protocol stacks is dual routing protocols. Building a combination APPN and multiprotocol network out of APPN-over-TCP/IP might look like the diagram shown in Figure 3. In this diagram, it is assumed that the APPN End Node-to-Network Node communication is accomplished using native SNA data link protocols and that only APPN NNs run on top of TCP/IP.

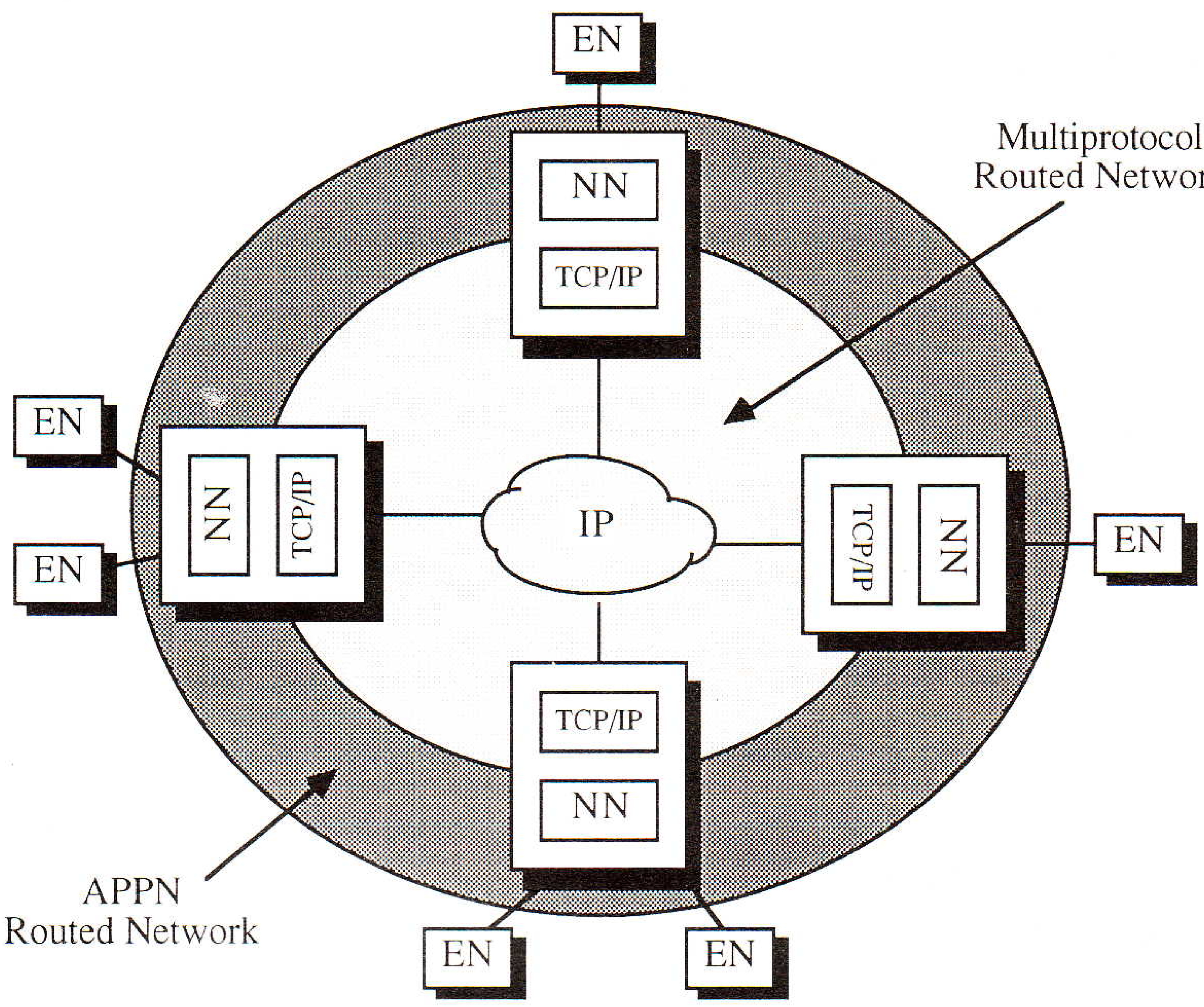

Figure 3: A Multiprotocol Network using APPN-over-TCP/IP

When APPN is placed directly on top of TCP/IP, the APPN routing protocol does not replace the routing protocol in the TCP/IP network—the routing protocol is merely enveloped in TCP/IP packets. The end result of this routing protocol layering is shown with the concentric circles in Figure 3. The routing protocol of the multiprotocol network, OSPF for example, has its view of the network topology that is consistent with the design of the IP network protocol while APPN has its own separate view of the network that is consistent with the design of the SNA protocol.

continued on next page

APPI *(continued)*

There is no exchange of network information between the two disjoint routing protocols. To the contrary, APPN's *Topology Database Updates* (TDUs) that reflect APPN's view of the network flow through the multiprotocol network encapsulated inside TCP/IP packets. This means that the IP network is carrying the routing protocol overhead of two networks:

- For example, OSPF's LSPs in IP packets in native fashion, and
- APPN's TDUs in encapsulated TCP/IP packets.

An APPI network uses a single routing protocol—whichever routing protocol happens to already be in effect for the multiprotocol network. Since ONNs provide APPN services to the End Nodes, SNA nodes are kept on the fringes of the multiprotocol network and a single routing protocol prevails.

An APPI network comparable to the one in Figure 3 above is illustrated below in Figure 4. Since there are no APPN Network Nodes in an APPI network, there are no APPN TDUs flowing though the multiprotocol network. Routers in the network all have a consistent view of the topology since it is managed with a single routing protocol.

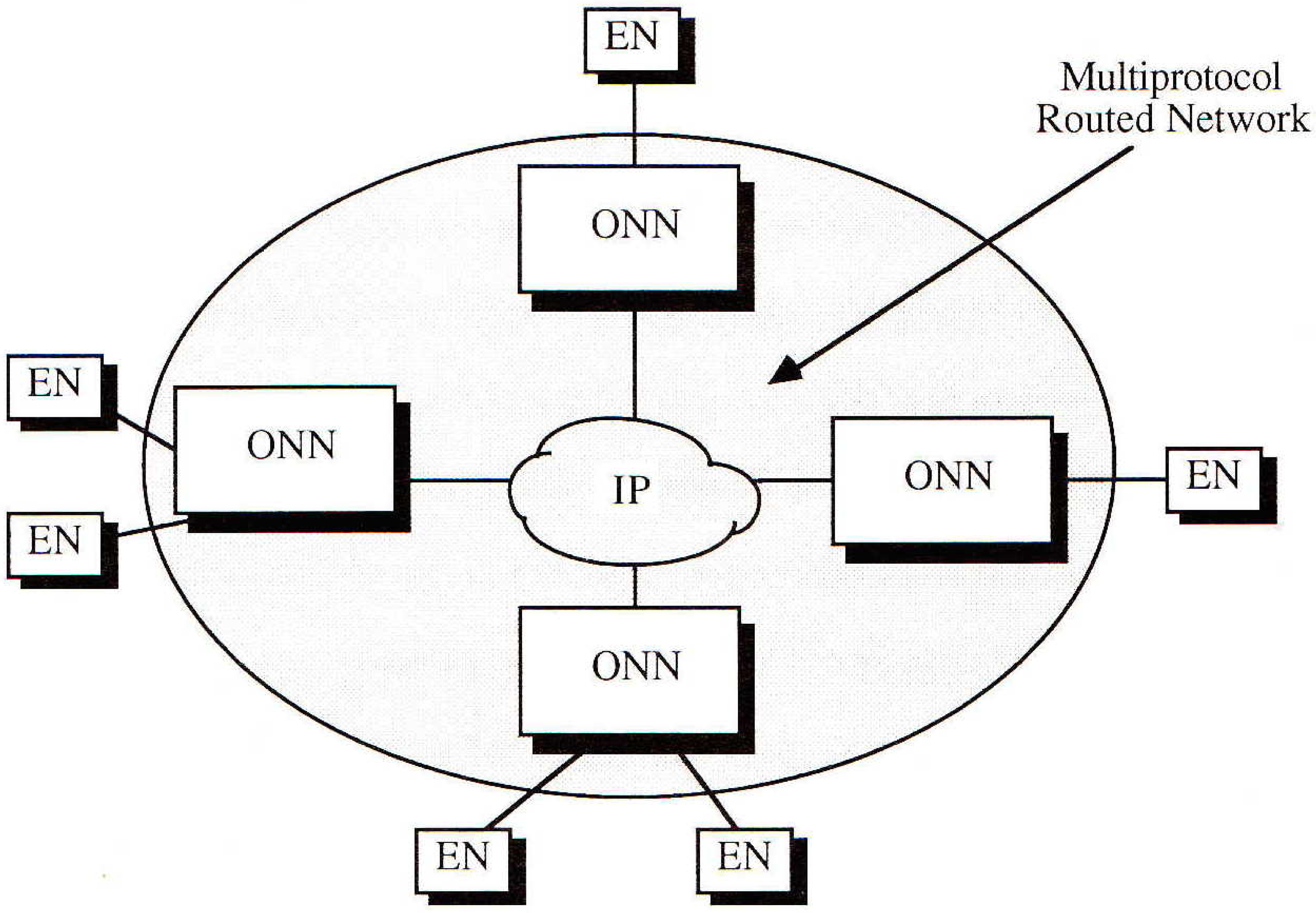

Figure 4: A Multiprotocol Network using APPI

Any activity to converge disparate routing protocols and thereby provide a single integrated routing protocol will occur within the realm of an open forum. Since the APPN routing protocol is closed and under the control of IBM, there is virtually no chance of APPN's routing protocol becoming an integrated solution in a multiprotocol world.

To the contrary, APPI accommodates SNA-based APPN End Nodes while still using open routing protocols to control the network. The way that APPI is designed, it can immediately take advantage of an integrated routing protocol while still permitting the SNA end devices access to the multiprotocol network. To the defense of APPN, there are several attributes of APPN (such as class of service support) that the multiprotocol world could benefit from.

Summary

This article represents the emerging architecture of APPI and discusses the pros and cons of APPI vs. APPN. The next meeting of the APPI Forum will occur at INTEROP 93 Spring in Washington, DC this week.

References

[1] "Networks of Small Systems," IBM Publication No. GG66-0216-00, July 1985.

[2] "APPN Architecture and Product Implementations Tutorial," IBM Publication No. GG24-3669-01, June, 1992.

[3] Joyce, Steven T. and Walker II, John Q., "Advanced Peer-to-Peer Networking (APPN): An Overview," *ConneXions*, Volume 6, No. 10, October 1992.

[4] Clark, Wayne, "SNA Internetworking," *ConneXions*, Volume 6, No. 3, March 1992.

[5] *ConneXions*, Volume 3, No. 8, August 1989, "Special Issue: Internet Routing."

[6] *ConneXions*, Volume 5, No. 1, January 1991, "Special Issue: Inter-domain Routing."

[7] Mockapetris, P., "Introducing Domains," *ConneXions*, Volume 1, No. 6, October 1987.

[8] Mockapetris, P., "Domain Names: Concepts and Facilities," RFC 1034, 1987.

[9] *ConneXions*, Special Issue: Network Management and Network Security, Volume 4, No. 8, August 1990.

[10] *ConneXions*, Special Issue: Network Management, Volume 3, No. 3, March 1989.

[11] Marshall T. Rose, *The Simple Book—An Introduction to Management of TCP/IP-based internets*, Prentice-Hall, 1990, ISBN 0-13-812611-9.

[12] *Interconnections: Bridges and Routers*, by Radia Perlman, Addison-Wesley, 1992, ISBN 0-201-56332-0.

WAYNE CLARK is the SNA Architect in the Engineering Department at Cisco Systems Inc. in Menlo Park, California. He has been working in the computer industry for 20 years and has worked on SNA implementations at Memorex, Ungermann-Bass, Communications Solutions Inc. (later part of 3Com), and Novell. His specialty is the implementation of LU 6.2 and NT 2.1 protocols. He was also the instructor of Interop's very first SNA tutorial in 1991. He has an M.S. in Computer Engineering from Santa Clara University. He is the acting technical director of the APPI Forum. He can be reached on the Internet as: `wclark@cisco.com`.

Write to *ConneXions*!

Have a question about your subscription? Are you moving, and need to give us your new address? Suggestions for topics? Want to write an article? A letter to the Editor? Have a question for an author? Need a *ConneXions* binder? Want to enquire about back issues? (there are now more than 72 to choose from and over 2000 total published pages; ask for our free 1987–1992 index booklet). We want to hear from you. Contact us at:

ConneXions—The Interoperability Report
480 San Antonio Road, Suite 100
Mountain View, CA 94040–1219
USA
Phone: +1 415-941-3399 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-949-1779
E-mail: `connexions@interop.com`

The OSF Distributed Computing Environment (DCE)

by David Chappell, Chappell & Associates

Introduction

In theory, distributed applications offer a number of advantages over their traditional single-system fellows. Yet actually building those applications can be quite challenging. Distributed applications require supporting services, things like a protocol allowing the application's various parts to communicate, a directory service, and mechanisms for providing security. It's not uncommon today for each distributed application to use custom-built supporting services, created just for that application. Doing this makes about as much sense as writing single-system applications each with its own custom operating system, i.e., in most cases, it makes no sense at all. A better approach is to factor out the common support services required for distributed applications into one common infrastructure, then build those applications on top.

This is exactly the approach taken by the Open Software Foundation's *Distributed Computing Environment* (DCE). With promised support from most major vendors, DCE is a vendor-neutral platform for supporting distributed applications. Among the services it provides are support for Remote Procedure Calls (RPCs), a directory service, security services, and a distributed file system. Taken together, these services provide something analogous to an operating system for distributed applications.

How DCE was created

The Open Software Foundation (OSF) is a membership organization devoted to the creation of vendor-neutral software infrastructures. In general, OSF doesn't create technologies from scratch. Instead, it takes advantage of work already done by vendors, universities, and anyone else active in an area of interest. For DCE, as for its other technologies, OSF issued a *Request For Technology* (RFT). An RFT is a brief document describing what problems OSF wishes to solve and soliciting solutions for those problems. For DCE, the problems included a mechanism for communication between parts of a distributed application, a way for these parts to find each other, security services for the application, and several more things. The RFT was widely distributed and anyone, OSF member or not, could respond.

The response to an RFT is not just a description of a proposed solution, however. To be a contender, actual working code must ultimately be provided, typically from a product the submitter already sells. OSF employees and a team of outside experts evaluated the submissions for each area, then selected those that they believed provided the best solutions. The code comprising those solutions, all written in **C**, was then integrated into a coherent whole by OSF and made available to the world. Anyone can license the DCE source code, but most licensees are system vendors and large software vendors. End users will typically buy shrink-wrapped DCE products from these vendors rather than license the DCE source directly from OSF (of course, some of the money vendors receive for these DCE products is paid to OSF as royalties, and some of *that* money is paid as royalties to the original creators of the technologies).

DCE 1.0 was released by OSF in January 1992. Early products based on this code began to appear by the end of that year, with a larger set of more complete DCE products available in 1993. Actual distributed applications based on DCE can be expected in sizable numbers shortly.

DCE components

DCE is not a simple thing, comprising as it does somewhere over one million lines of source code. It consists of several different components, each of which depend to a greater or lesser extent on the others. Perhaps the most fundamental of those components is DCE's mechanism for *Remote Procedure Call* (RPC).

RPC

Traditional applications written in a high-level language almost universally are broken into a series of *procedures* (also called *subroutines* or *functions*). Each of these procedures generally takes some number and type of parameters, performs some function, and returns some kind of result. When making the jump to distributed applications, starting with this familiar procedural model is very attractive. A fairly obvious extension is to separate the caller of the procedure from the procedure body, to let the first execute on one machine and the second on another. To the caller, these procedures behave much the same as ordinary procedures. Their remoteness, and the fact that they may actually execute on some other system entirely, is hidden.

RPC fits very naturally into a client–server model. The process that invokes a remote procedure is the client, requesting some service, while the process that executes that procedure is the server. RPC is not a new idea, and DCE's RPC component is derived from the *Network Computing System* developed by Apollo (now part of Hewlett-Packard).

IDL

While RPC is an undeniably elegant idea, actually implementing it requires some work. For example, client and server must agree on exactly how each remote procedure should be invoked. In DCE RPC, this agreement is embodied in an *interface.* Each interface contains the definitions for one or more procedures and their parameters expressed in DCE's *Interface Definition Language* (IDL). For a simple picture of IDL, imagine ANSI **C** with only function prototypes, typedefs, and constant definitions (there are in fact several important extensions, but these things constitute the bulk of the language). An example interface is shown in Figure 1.

```
[ uuid(A01D0280-2D27-11C9-9FD3-08002B0ECEF1),
  version(1.0) ]

interface math {
        const long ARRAY_SIZE 10;
        typedef long array_type[ARRAY_SIZE];
        long get_sum([in] long first,
                     [in] long second);
        void get_sums([in] array_type a,
                      [in] array_type b,
                      [out] array_type c);
}
```

Figure 1: An IDL Interface

The unusual 16 byte hex value shown at the beginning of Figure 1's interface is a *Universal Unique Identifier* (UUID). As their name implies, UUIDs are globally unique, and they can be generated at will by any DCE system (uniqueness is ensured by incorporating a timestamp and a unique system identifier in each UUID value). Every interface is assigned a UUID to distinguish it from all others. UUIDs are also used for other things in DCE, including providing a unique identifier for every human user.

Following the UUID are a version number for the interface and the constants, types, and procedures that interface defines. The syntax for each of these definitions is derived from ANSI **C**, with the most obvious addition the [in] and [out] preceding each parameter.

continued on next page

The OSF DCE *(continued)*

These attributes indicate whether values are passed into or out of the procedure (it's also possible to say [in, out]) and thus control exactly what information is copied when during the remote procedure's execution.

Mechanism

When a client process invokes a remote procedure, the procedure's body is (obviously) not actually present in that process. Still, the client must invoke something when it makes the call. What is actually invoked is called the client *stub*. Among other duties, this stub converts the procedure's parameters into a form suitable for transmission across the network (a process called *marshalling*) and causes one or more packets to be sent to the server. At the server, a server stub *unmarshalls* the parameters, putting them in the correct format for that system, and invokes the requested procedure. The procedure executes, then sends any results back via the same path, i.e., through the stubs. Once complete, the called procedure returns from the client stub back to its caller like an ordinary procedure. Throughout this process, the applications and the stubs rely on library routines linked into both the client and server process. Known collectively as the RPC *runtime,* these routines provide a number of basic services. The path taken by an RPC is shown in Figure 2.

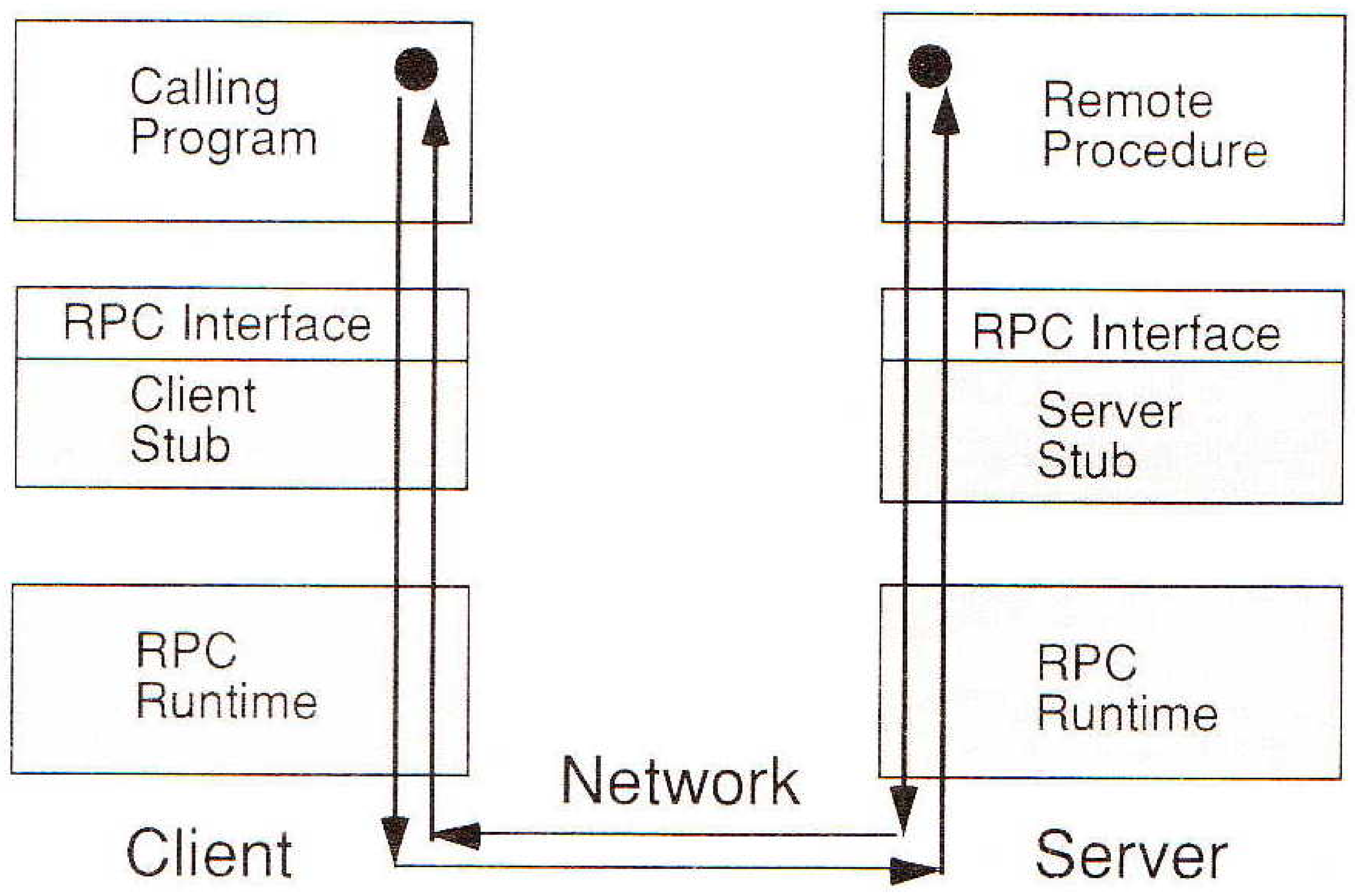

Figure 2: RPC Operation

While masochists might choose to write these stubs themselves, DCE provides a more convenient option with the IDL compiler. Taking an IDL-defined interface as input, the IDL compiler generates appropriate client and server stubs for that interface, along with a header file to be incorporated by the users of those stubs. The generated stub code is then linked with the user-written client and server code and the runtime library to create a complete application. This process is illustrated in Figure 3.

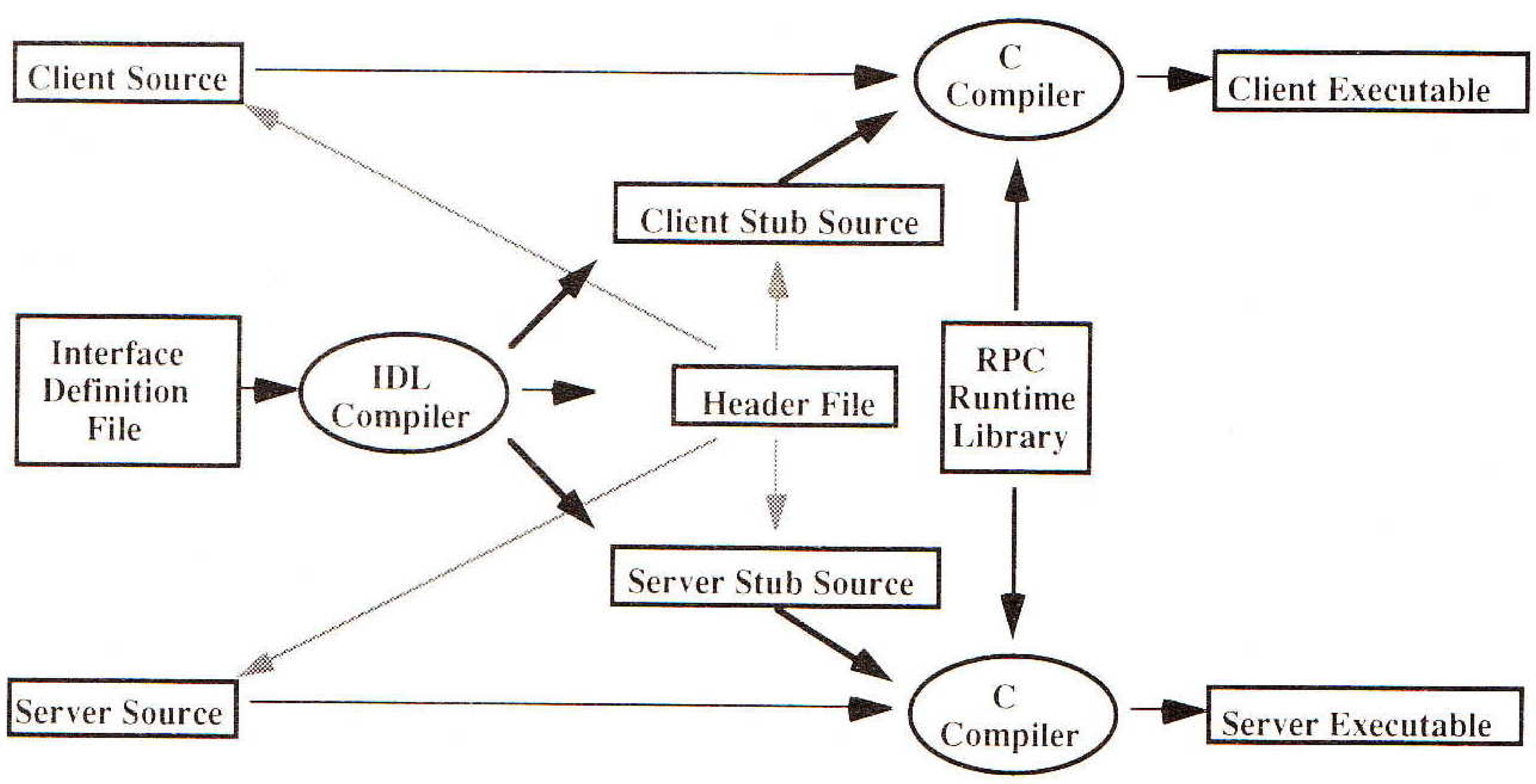

Figure 3: Creating an RPC Application

Creating a distributed application with DCE, then, requires specifying all interactions between clients and servers in one or more IDL interfaces. These interfaces are used to produce stubs, which in turn are combined with the actual application code. Whatever the distributed application is doing, DCE RPC can provide a foundation for interaction between its components.

An aside: cells

In most distributed environments, most clients perform most of their communications with only a small set of servers. This locality of reference is made explicit in DCE through the notion of a *cell.* A cell consists of some number of clients and servers (along with the machines on which they run) that do most of their communication with each other. A cell's size both in number of machines and in geographical extent is determined by the people administering the cell—there are no fixed limits. Although DCE allows communication between clients and servers in different cells, it optimizes for the more common case of intra-cell communication.

Threads

There's a fairly obvious limitation with RPC as a mechanism for supporting distributed applications. The limitation is this: since each call acts like an ordinary local procedure, the caller is blocked until the procedure completes. For non-remote procedures, this probably makes sense, since caller and callee both run inside the same process on the same system. For a remote call, however, there are generally at least two CPUs involved, one at each system. Why block the entire calling process waiting for a remote server to complete? Similarly, why block other clients while a server executes a procedure for only one of them? To make effective use of the parallelism inherent in a distributed environment, some way around this blockage must be found.

DCE's solution to this problem is to incorporate *threads.* The basic idea is straightforward, and is illustrated in Figure 4: allow a single process to have more than one simultaneous flow of control. For clients, this means that while each call to a remote procedure will still block, it will block only its own thread; other threads in the process are free to continue executing. For servers, support for threads means that a single server process can service requests from multiple clients at once, rather than forcing one client to wait until all preceding requests have been completed.

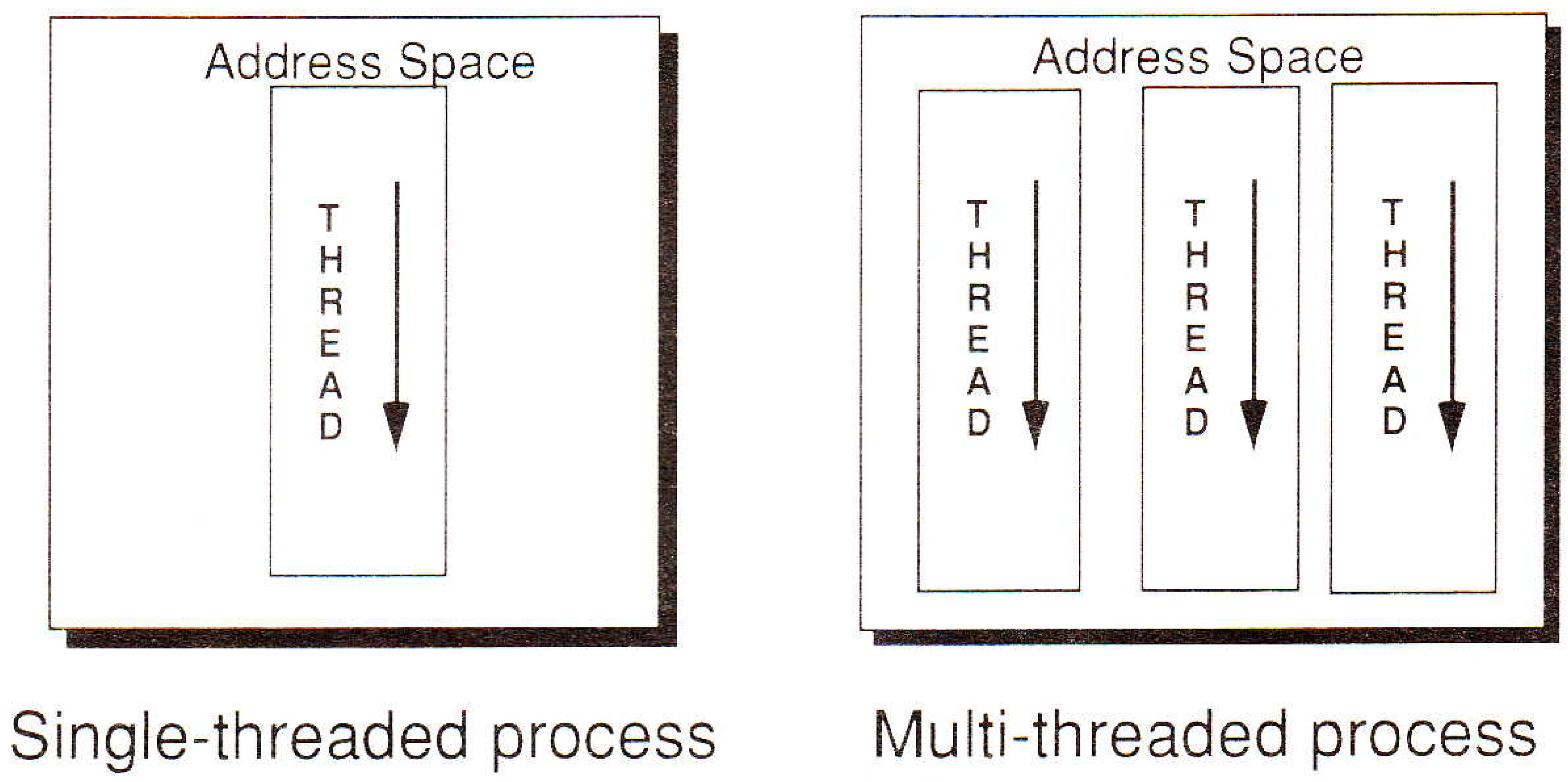

Figure 4: Threads

Some operating systems provide direct support for multi-threaded processes, while others (probably the majority today) do not. DCE specifies an application programming interface (API) to access thread services. Called *pthreads,* it is a slight extension to work done by an IEEE POSIX committee.

continued on next page

The OSF DCE *(continued)*

If a system supporting DCE has operating system kernel support for threads, the pthreads interface can serve as an API to it. If a system supporting DCE has no intrinsic support for threads, the pthreads API can interface with a threads library linked into the process. This library provides all of the functionality required to control threads within that process. In this situation, the operating system is unaware that the process is multi-threaded. To a programmer, however, things look the same whether or not kernel threads are supported.

The pthreads interface provides routines that allow a programmer to create and terminate threads, have one thread wait for another to complete, and perform various kinds of thread synchronization. In a typical client, a separate thread might be started for each remote procedure call, allowing several to be in progress at a time. If one thread needs the result of another's RPC before continuing, it can simply wait for that thread to complete, then proceed. While this style of programming adds complexity to an application developer's life, it also makes RPC much more useful.

Directory Services

Dividing applications into clients and servers is all very well, but how exactly do those clients find appropriate servers? Solving this problem is the job of DCE's *Directory Service.* To access a server, a client needs to acquire that server's *binding information,* information that the server must first place into the directory. Making RPCs to the server requires only that the client learn the server's name, then use this name to look up the binding information. While the DCE directory service can be used for more general purposes, its most common application at the moment is this simple but critical function.

DCE's designers reasoned that most client searches for binding information would be for servers in the same cell. Intra-cell lookup, then, should be as efficient as possible. There is also some advantage in a directory service that is entirely under OSF's control, i.e., one that isn't a complex, rigid standardized solution. To meet these requirements, DCE includes a *Cell Directory Service* (CDS). CDS is derived from Digital Equipment Corporation's *DECdns* directory service (although a number of modifications were made to the original technology, including making it run over DCE RPC and using DCE security). When a server wishes to make its binding information available to clients, it *exports* that information to one of its cell's CDS servers. When a client wishes to locate a server within its own cell, it *imports* that information from the appropriate CDS server. A client actually performs this operation by calling on a CDS *clerk,* a process resident in the client's system.

Sometimes, though, clients need to access servers in foreign cells, i.e., in cells other than their own. For this to work, the CDS servers in all cells must somehow be linked together. An obvious protocol contender for carrying out this linkage is CDS itself. But since DCE is envisioned as eventually comprising a large number of cells scattered all over the world, this would require maintenance of a worldwide CDS directory of some sort, an unappealing task. Instead, DCE's designers chose to use an existing global directory to link cells together: the *Domain Name System* (DNS), already in very wide use on the Internet. Each cell is assigned a domain name (e.g., `osf.com`), and information about how to find a CDS server in that cell is stored under that name in DNS. To access a server in a foreign cell, a client gives the cell's name and the name of the desired server within that cell.

A component called a *Global Directory Agent* (GDA) extracts the location of the named cell's CDS server from DNS, then a query is sent directly to this foreign server. The components of the DCE directory service and how they interact are shown in Figure 5. (One important note: DCE officially allows the use of OSI's X.500 in place of DNS. In this case, cells are given X.500-style names. Even though DCE actually includes X.500 source code, it seems unlikely that it will be used much for linking cells until there exists a worldwide directory system that runs X.500. When, or perhaps if, that happens, DCE is ready.)

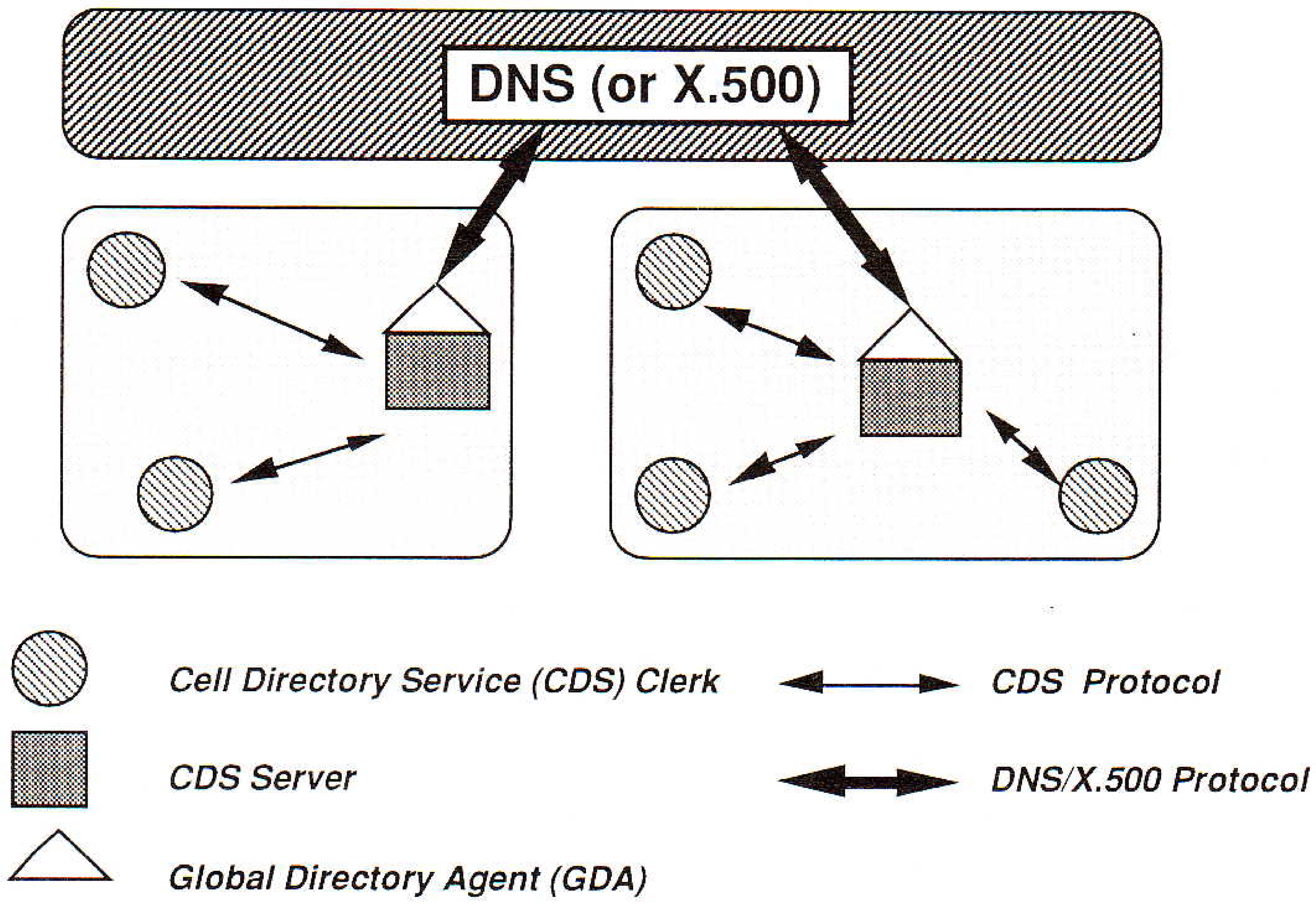

Figure 5: DCE Directory Components

Namespace

While this multi-part structure makes excellent technical and administrative sense, it leads to some complexities in actually naming things. The DCE namespace is conceived as a single, worldwide structure, with a global root denoted by the symbol "/ . . .". Below this root appears (for most cells, anyway) the DNS namespace, used to name each cell. And finally, each cell contains its own internal namespace, starting from the cell root. An example of this structure appears in Figure 6.

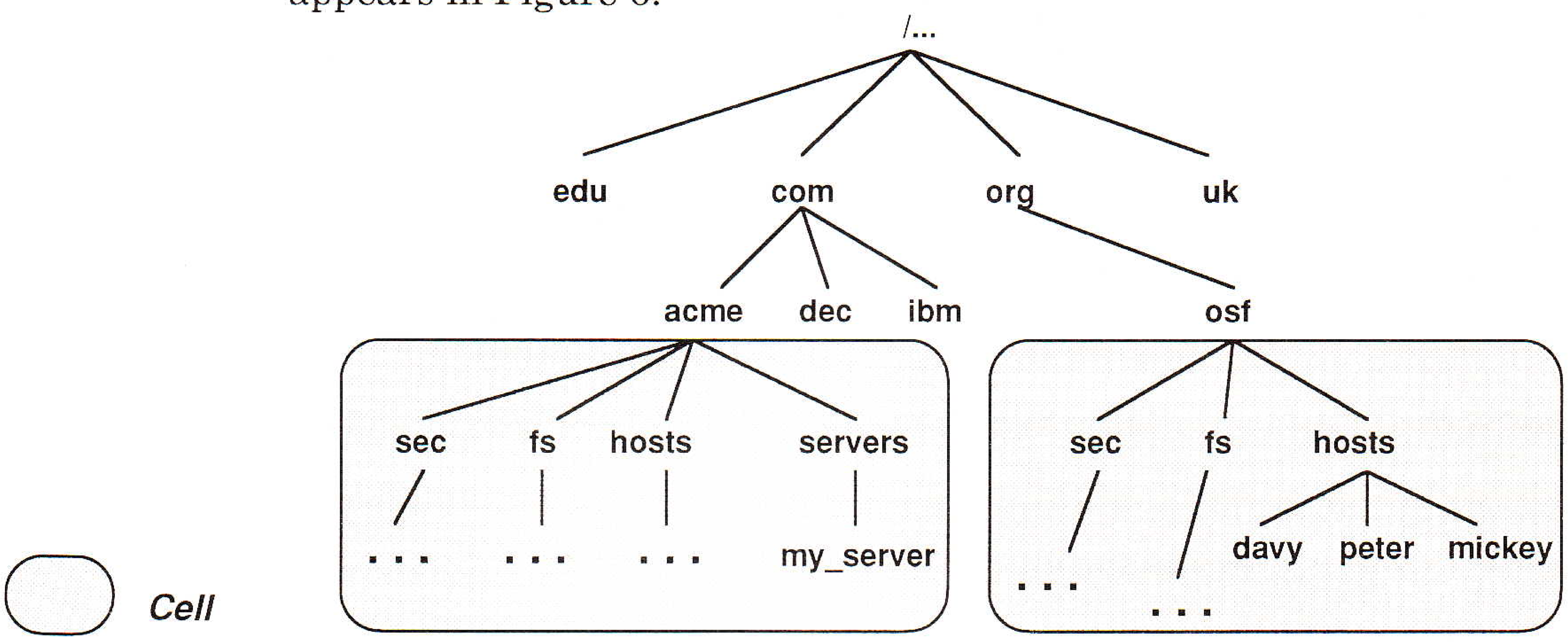

Figure 6: The DCE Namespace

As is shown in the figure, each cell typically contains a number of standard directories, including:

- *sec:* information about each user in the cell is contained here. Although part of the DCE namespace, this information is not actually maintained by CDS. Instead, it's kept in the security service, described next.

continued on next page

The OSF DCE *(continued)*

- *fs:* all files managed by DCE's *Distributed File Service* (DFS) have names below this point in the cell namespace. Once again, although these names are part of the DCE namespace, the named items (i.e., the files) are not maintained by CDS. Instead, they are stored on and accessed through DFS, as described later.
- *hosts:* each machine that belongs to the cell has an entry here (and this information actually is maintained by CDS). In the figure, the cell `osf.org` contains machines named `davy`, `peter`, and `mickey`.

Each of the components in a DCE name is separated with a slash. For example, a server in the first cell shown in the figure might have the global name of `/.../acme.com/servers/my_server`, whereas `/.../osf.org/hosts/mickey` names a particular host in that cell. These names identify the same things from anywhere in the DCE world. To make life easier for people who may be typing these names, a shorthand form exists for use only within the server's cell. If a name begins with "`/.:`" rather than "`/...`", it is only a cell-relative name rather than a global one. A client in the same cell as the server named above could refer to it simply as `/.:/servers/my_server`.The symbol "`/.:`" is really shorthand for "the root of the local cell."

Security

Virtually no one today would use a multi-user operating system that did not include effective security measures. The same caution ought to apply to a platform for building distributed applications. With this in mind, DCE's designers have incorporated a number of security services. There are four major services critical to providing secure distributed applications:

- *Authentication:* in short, authentication means proving you are who you say you are. When a client requests some service from a server, it must not only identify itself, it must provide some information that proves that this is its true identity. When a human user logs into a computer system, she commonly types her login name and password. The password, known only to her, verifies that this is her true identity. Solving the same problem between the parts of a distributed application is substantially more complex, but no less important.
- *Authorization:* once a server has authenticated a client, the next question to be answered is this: does this client have the right to perform the service it's requesting? The request may be to invoke a particular remote procedure or to access a certain file or to modify an entry in CDS or to perform any other service this server makes available. Whatever it is, an authorization decision must be made. Authorization is sometimes called "access control," perhaps a slightly more descriptive name since access is exactly what is being controlled.
- *Data integrity:* when data is sent across a network there exists the potential for someone to modify that data during transit. For example, a request from client to server to add $100 to a checking account might be intercepted, changed to $100,000, then sent on its way. The service of data integrity guards against this by allowing the recipient of a message to determine whether it has been tampered with.
- *Data privacy:* this is perhaps the most obvious of the security services provided by DCE. It ensures that data sent between clients and servers cannot be read by anyone but the parties involved in the communication.

Security services are actually provided by security *mechanisms.* In DCE, the services of authentication, data integrity, and data privacy are provided by a slightly modified version of MIT's *Kerberos* Version 5. For authentication, Kerberos issues a ticket to clients that allows them to prove their identity to servers. For data integrity, a checksum is computed for each packet sent, then encrypted using a key known only to the client and server. Any modification to the data will be detected, since the receiver's calculation of the checksum will not match the decryption of the checksum value it received. And for data privacy, the data is simply encrypted before transmission, again using a key known only to the client and server (this key is provided by Kerberos along with the ticket). All of this encryption is done with an encryption algorithm called the *Data Encryption Standard* (DES).

Kerberos by design does not address the problem of authorization. DCE does, however, and it uses one common mechanism to provide this service: *Access Control Lists* (ACLs). An ACL is exactly what it sounds like: a list that controls who can access some service or object. When a server receives a request, that request typically contains the *Privilege Attribute Certificate* (PAC) of the requestor. This PAC identifies who made the request and what groups he belongs to (both the requester's identity and his groups are denoted by UUIDs). A component of the server called the *ACL Manager* compares the UUIDs from the PAC with the entries on the ACL of the desired object. If they match correctly, access is allowed. If not, access is denied.

Application developers can select which, if any, of these services they wish to use via the RPC API. Since adding security can subtract from performance, DCE's designers did not wish to mandate any particular level of service. By providing security, however, DCE becomes a candidate platform for mission-critical applications, something that likely would not be true if these services were omitted.

Distributed Time Service

Computers typically come with built-in clocks. These clocks aren't always especially accurate, and they're often set in a casual way (e.g., by an administrator glancing at her watch and setting an approximate time). For an effective distributed environment, clocks on all systems must be fairly closely synchronized. In DCE, this is accomplished using the *Distributed Time Service* (DTS).

Derived from a DEC-defined protocol, DTS exhibits the usual client–server structure. DTS clients request the correct time from some number of servers (typically three), then reset their clocks as necessary to reflect this new knowledge. How often a client resynchronizes, and thus how accurate its clock will be, is configurable by the system's administrator.

There's an obvious problem with this scenario: when a server responds to a client's request, it checks its own clock to determine the time value to send back. By the time the packet containing this time arrives at the client, it's no longer correct because of the inherent delay in packet delivery. To handle this problem, the client attempts to compensate for the round-trip time of its request (actually an RPC) to the server. In addition, DTS does not define time as a single value. Instead, a client querying a server for the correct time receives in return an interval, expressed as a time plus or minus an inaccuracy. A client computes a new interval that is the intersection of the intervals it receives from servers, then sets its own clock to the midpoint of this interval. Servers synchronize with each other in a similar way.

continued on next page

The OSF DCE *(continued)*

Ideally, one server has access to a source of high-quality time, e.g., a radio listening to a continual time broadcast or a *Network Time Protocol* (NTP) connection from the Internet. If this is true, DTS will keep the machines in a cell in sync both with each other and with the true time.

Distributed File Service

Everything discussed so far is part of DCE's basic platform for supporting distributed applications. DCE itself actually includes something that can be viewed as just such an application, however, one that makes use of all of the services just described. DCE's *Distributed File Service* (DFS) allows sharing of files among clients and servers in the same cell or in different cells. Derived from the *Andrew File System* (AFS), it is built on DCE RPC, uses threads to enhance parallelism, relies on the DCE directory to locate servers, and uses DCE security services to foil attackers.

DFS has a few important differences from DCE's other components. One of these is its optional nature: it is entirely possible to use the rest of DCE without DFS. This is not true of any of the components discussed above; creating a true DCE cell requires using all of them. Also, DFS is the only DCE component that is strongly biased toward UNIX. Although likely to be available on a variety of operating systems, its view of files and file semantics is exactly that taken by UNIX.

Like everything else in DCE, DFS relies on a client–server architecture. Clients, called *cache managers,* communicate with servers using RPC. The slightly unusual name for clients derives from the way they work, which is heavily dependent on caching. When a file is first accessed, a client copies the file's first *chunk* back to its local disk. The default chunk size is 64Kbytes, so many (perhaps most) files will be copied in their entirety. A client is then free to read and write the data in its cache. A primary reason for taking this approach is obvious: better performance. Once a chunk is present in the cache, access takes place without the delays inherent in access across a network. If multiple clients are reading and writing the same chunk of the same file, a somewhat elaborate token mechanism is used to maintain consistency among their caches, ensuring that each client sees the most current copy of the data.

DFS is designed to support a group of servers within a cell that provide file access for that cell's clients. Toward this end, DFS presents a single file namespace to its clients. As shown in Figure 7, all of the files managed by DFS in a cell appear within the *fs* directory immediately below the cell root. From within a cell, a DFS file can be specified as, for example, `/.:/fs/tmp/test` or, using a DFS-specific shorthand notation, as just `/:/tmp/test`. From a foreign cell, the same file could be referenced as `/.../acme.com/fs/tmp/test`, a name that uniquely identifies this file from anywhere in the world. A group of interconnected DCE cells supporting DFS can provide a global file system with files accessible to any of their clients from anywhere in the world.

DFS defines how clients talk to servers. How a server actually accesses its files is defined by the server's file system. To get the full functionality of DFS requires that the server use DCE's *Local File System* (LFS), also known as *Episode.* DFS servers with LFS allow file replication, support standard DCE ACLs, and provide various administrative conveniences. Although DFS is certainly useful without LFS, this extra component adds significantly to the service it provides.

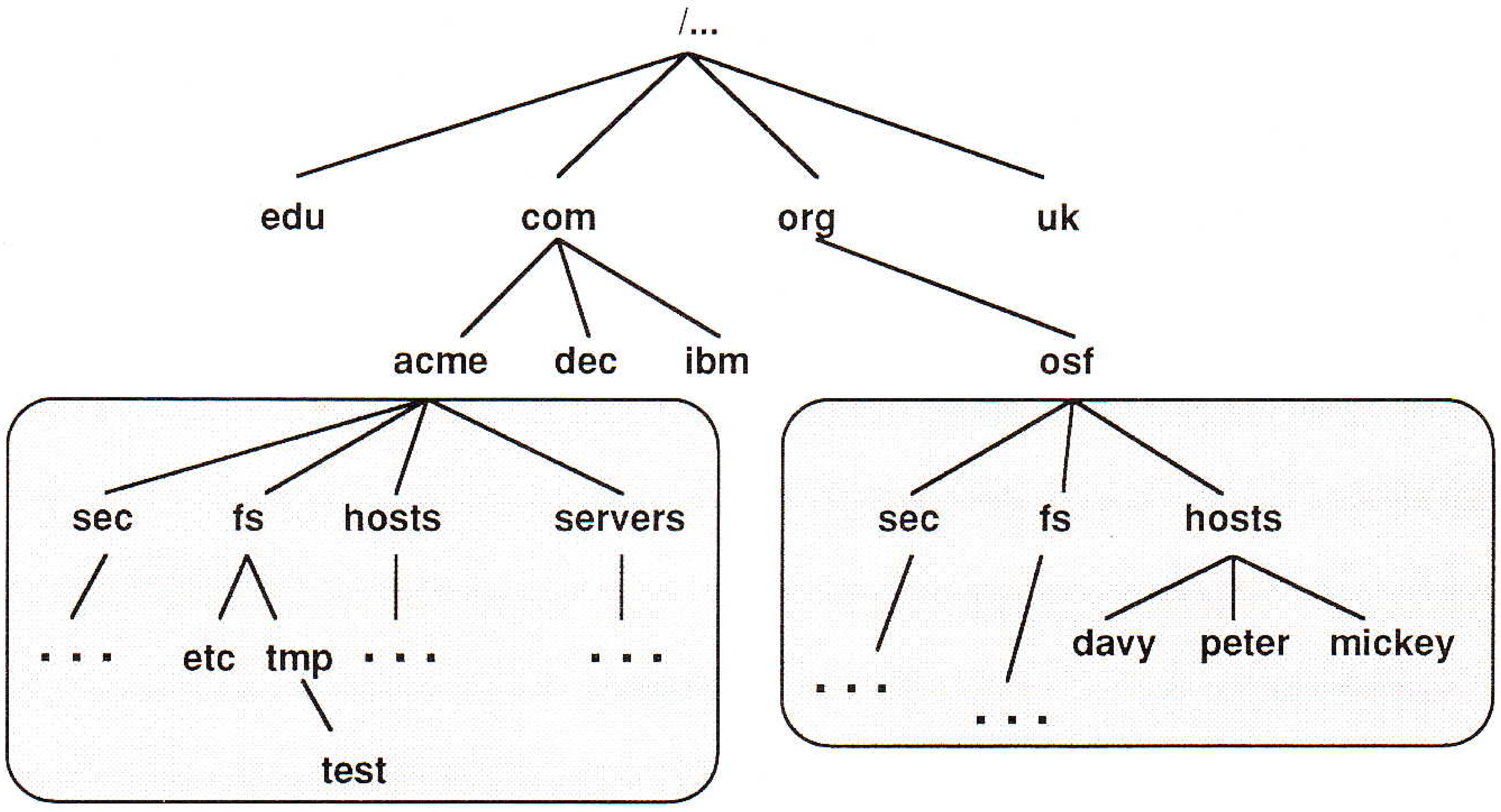

Figure 7: DFS Files in the DCE Namespace

Conclusion

DCE provides a common base for supporting distributed applications. It is not simple, but it is also not too complex. DCE is positioned to become the standard platform for supporting distributed applications in a multi-vendor environment. It's still a bit too soon to know whether this will actually come to pass, but the prospects for its success are undeniably quite good.

References

[1] Rosenberry, Ward, David Kenney, and Gerry Fisher, *Understanding DCE,* O'Reilly & Associates, 1992, ISBN 1-56592-005-8. *An overview of DCE, this book provides a broad introduction to the topic. The second half of the book gives a high-level look at planning and setting up a DCE cell, an especially useful contribution.*

[2] Shirley, John, *Guide to Writing DCE Applications,* O'Reilly & Associates, 1992, ISBN 1-56592-004-X. *DCE's main purpose is to provide a platform for distributed applications, and this book describes in some detail how to actually write them. Although it omits some important topics (like threads and security), it's still an excellent introduction to the area.*

[3] Millikin, M., "Distributed Computing Environments: Laying the foundation for the future of interoperability," *ConneXions,* Volume 4, No. 10, October 1990.

[4] Millikin, M., "Networks and Objects," *ConneXions,* Volume 5, No. 10, October 1991.

[5] Chappell, D. "The OSF Distributed Management Environment," *ConneXions,* Volume 6, No. 10, October 1992.

DAVID CHAPPELL is principal of Chappell & Associates, a training and consulting firm focused on distributed computing for open systems, and the instructor for Interop's DCE/DME tutorial. This article is based on his chapter from the forthcoming book *Distributed Computing: Implementation and Management Strategies,* edited by Raman Khanna and scheduled for publication by Prentice-Hall this fall.

Internet Talk Radio

by Carl Malamud

Introduction

Over the past few years, two trends have come together to present an opportunity for a new type of journalism. On the one hand, the trade press has focused on marketing and product reviews, leaving an ever-larger gap for a general-interest, technically-oriented publication focused on the Internet. At the same time, the Internet has made great progress in supporting multimedia communication, through standards such as IP multicasting and MIME messaging.

Internet Talk Radio attempts to fuse these two trends and form a new type of publication: a news and information service about the Internet, distributed on the Internet. Internet Talk Radio is modeled on National Public Radio (NPR) and has a goal of providing in-depth technical information to the Internet community. The service is made initially possible with support from Sun Microsystems and O'Reilly & Associates. Our goal is to provide a self-sufficient, financially viable public news service for the Internet community.

Flame of the Internet

The product of Internet Talk Radio is an *audio file,* professionally produced and freely available on computer networks. To produce these files, we start with the raw data of any journalistic endeavor: speeches, conference presentations, interviews, and essays.

This raw information is taped using professional-quality microphones, mixers, and DAT recorders. The information is then brought back to our studios, and edited and mixed with music, voice overs, and the other elements of a radio program. The "look and feel" we strive for is akin to NPR's "All Things Considered" or other programs that appeal to the general interest of the intelligent listener.

Our goal is to hit the topics that don't make it into the trade press. Instead of SNMP-compliant product announcements, we want to present descriptions of SNMP. Instead of articles on GOSIP, we want to describe the latest Internet Drafts and place them in perspective. Instead of executive promotions, we want to give summaries of mailing list activity and network stability. Instead of COMDEX, we want to cover the Internet Engineering Task Force (IETF).

Town Crier to the Global Village

The result of Internet Talk Radio's journalistic activities is a series of audio files. The native format we start with is the Sun Microsystems `.au` format, closely related to the NeXT `.snd` format. This format consists of the CCITT *Pulse Code Modulation* (PCM) standard of 8 bits per sample and a sampling rate of 8000 samples per second, using the µ-law encoding (a logarithmic encoding of 8 bit data equivalent to a 14 bit linear encoding). A half-hour program would thus consist of 64,000 bits per second or 15Mbytes total.

Programs are initially spooled on UUNET, the central machines of the Alternet network. Files are then moved over to various regional nets for further distribution. For example, EUnet, a commercial network provider for Europe with service in 24 countries, will act as the central spooling area for the European region. The *Internet Initiative Japan* company will provide the same service for Japanese networks.

The goal of coordinated distribution is to reduce the load on key links of the network. Transferring a 15Mbyte file over a 64Kbps link does not make sense during peak times. On the other hand, a leased line has the attribute that a bit unused is a bit forever gone. Transferring large files at low priority in non-peak times has little or no incremental cost.

Files thus move from the UUNET central spool area, to regional spools, to national and local networks. We anticipate most of this transfer to be done using FTP, but some networks are discussing the use of NNTP news groups and MIME-based distribution lists.

Distribution

It is important to note that Internet Talk Radio is the source of programming and does not control the distribution. These files are publicly available, subject only to the simple license restrictions of no derivative work and no commercial resale.

Distribution is controlled, as with all other data, by the individual networks that make up the Internet. We intend to work closely with networks all over the world to ensure that there is some coordination of distribution activity, but ultimate control over this data is in the hands of those people who finance, manage, and use networks.

We don't believe indiscriminate use of anonymous FTP is the proper method for distributing large archives. Previous experience with ITU standards, RFC repositories, and with large software archives such as The X Window System indicates that setting up a top-level distribution hierarchy goes a long way towards alleviating network load.

Even with a top-level hierarchy, however, there will always be anonymous FTP sites and there will always be people that go to the wrong FTP server. This behavior is largely mitigated by setting up enough "local" servers and publicizing their existence. Like any large distributor of data, we are mindful of the load on the transcontinental and regional infrastructures and will take aggressive steps to help minimize that load.

Asynchronous Times, Asynchronous Radio

Once files have made their way to a local or regional network, they are moved to the desktop and played. Once again the individual users of the network decide how to present data. We hope to see a wide variety of ways of having our files played and only list a few of the more obvious methods.

The simplest method to play an `.au` file on a SparcStation is to type "play filename." If the file is placed on a *Network File System* (NFS) on a central server, the user simply mounts the file system and plays the file. Alternatively, the user copies the file to a local disk and plays it.

More adventuresome playing of files uses *multicasting*. A simple multicast program called "radio" for a local Ethernet is available from CWI, the mathematics institute of the Netherlands. A more sophisticated approach, IP Multicasting, allows a program to reach far beyond the confines of the Ethernet.

IP multicasting might be used on a local basis, or can have a global reach. There is a consortium of regional networks that have formed the *Multicast Backbone* (MBONE), used for audio and video programming of key conferences such as the IETF.

Internet Talk Radio does not assume use of the MBONE for playing files. Needless to say, the operators of the MBONE are free to play Internet Talk Radio files (and we would be delighted if this happens), but it is up to the local network affiliates to determine how and when they distribute this audio data.

In many cases, people will want to play files on a wide variety of different platforms. The *Sound Exchange* (SOX) program is a publicly-available utility that easily transforms a file from one format to another. Using this utility, the Mac, Silicon Graphics, DECstation, PC, and many other platforms can play Internet Talk Radio files.

continued on next page

Internet Talk Radio *(continued)*

Geek of the Week

In the spirit of dignified, conservative programming, the first production from Internet Talk Radio is dubbed *Geek of the Week*. Geek of the Week features technical interviews with key personalities on the Internet. Some of the people who have agreed to appear on Geek of the Week include Daniel Karrenberg of the RIPE NCC, Dr. Marshall T. Rose of Dover Beach Consulting, Milo Medin of the NASA Science Internet, and Daniel Lynch of Interop Company.

Geek of the Week focuses on technical issues facing the Internet. This initial program is sponsored by Sun Microsystems and O'Reilly & Associates. Their support makes it possible for Geek of the Week to be produced professionally and then to be distributed at no charge.

AUP questions

One of the issues that Internet Talk Radio faces is the vestiges of *Appropriate Use Policies* (AUPs) that linger from the original ARPANET days. While Sun Microsystems and O'Reilly & Associates view Internet Talk Radio in terms of an investigation of on-line publishing, of multicasting, and other engineering issues, we feel it important that our sponsors are given due credit in the programs.

At first glance, this smacks of the crass and commercial. Indeed, it smacks of advertising. Jumping to that conclusion, however would be a simplistic mistake. The Appropriate Use Policies were formulated to guarantee that networks are used for the purposes envisioned by the funding agents. In the case of an AUP-constrained networks such as the NSFNET, this means that use of the network must benefit U.S. science and engineering.

We feel that an in-depth interview with Internet architects clearly falls within the purview of all AUP policies. However, we understand that certain networks may not accept certain types of programming. For this reason, our central spool areas are carefully picked so they are AUP-free. This way, if a network feels the programming is inappropriate, they can simply inform their users not to obtain or play the files.

It should be noted that one advantage of supporting the professional dissemination of news and information up-front is that the user is not directly charged. Somebody has to pay for information to be produced, and the sponsorship model means that copy protection, accounting, security, and all the other complications of a charging model are avoided and that high-quality news and information becomes increasingly available on the Internet.

The medium is the message

While Geek of the Week is our flagship program, we intend to intersperse mini-features throughout. *The Incidental Tourist,* for example, will feature restaurant reviews and other travel information for sites throughout the world. *The Internet Hall of Flame* will highlight nonlinear behavior on mailing lists, and we will have periodic book reviews by Dan Doernberg, one of the founders of Computer Literacy Bookshops.

The logical extension to Geek of the Week is to begin coverage of industry functions. To date, we have received permission to tape for later rebroadcast sessions and presentations at the European RIPE meetings, the IETF, and at the INTEROP Conferences. We are negotiating with other industry forums to try and establish permission to cover additional conferences. Our hope is to begin providing news summaries of these key events. If you can't make it to the IETF, for example, Internet Talk Radio would like to provide a half-hour news summary describing what happened on each day.

The next logical step is to begin producing analysis of key technical topics. Here, we look at in-depth (e.g., 15 minute) summaries of technical topics such as MIME, proposals for the next IP, SNMPv2, or the architecture of the *Global Internet Exchange* (GIX). We would also furnish analysis of political topics, such as the POISED effort to reorganize the Internet standards process, or the background of the IPv7 debate.

Eventually, our hope is to combine all these reports together and form a daily news broadcast to the Internet. When you walk in and start reading your mail, you simply click on the "radio" icon and listen to Geek of the Week while deleting messages from the more hyperactive mailing lists.

Tomorrow is the future

The "radio" metaphor was carefully chosen. We wanted an alternative to plain ASCII files, yet did not feel that the Internet infrastructure was ready for regular video feeds. Production of video or true multimedia required an order-of-magnitude higher investment in production facilities. After all, we know bad TV since we see so much of it.

Eventually, Internet Talk Radio wants to go beyond the confines of the simple radio metaphor. Already, we describe the service as "asynchronous radio," recognizing that our listeners can start, stop, rewind, or otherwise control the operation of the radio station.

As a multicasting infrastructure gets deployed throughout the Internet, we see the opportunity to expand the radio metaphor and begin the creation of a truly new news medium. Multicast groups and videoconferencing tools allow the creation of an *Internet Town Hall,* a moderated forum with a very wide reach or games shows like *Name That Acronym* where everybody gets to play.

Because we are on the Internet, we can add a wide variety of different programming techniques. While listening to a series of interviews about MIME messaging, for example, you might also scroll through a series of Gopher menus that hold more information about the MIME standards, or search a WAIS database for a biography of the speakers.

We hope that Internet Talk Radio will be the first of many such information services on the Internet, supplementing the random anarchy of news and mailing lists with professionally produced news and information. Indeed, we hope that Internet Talk Radio forms the first of many "desktop broadcasting" efforts.

Internet Talk Radio debuts at the Columbus IETF meeting at the end of March. Stay tuned for more information.

For more information

Guido van Rossum, "FAQ: Audio File Formats," available via anonymous FTP from `ftp.cwi.nl:/pub/AudioFormats2.10`. An excellent introduction to audio formats, encoding, and other information about sound files on different platforms. This same site also has copies of the *SoundExchange* (SOX) program for translating files into different audio formats, and the Radio program for playing a sound file on an Ethernet.

CARL MALAMUD is the author of several professional reference books, including *Stacks* and *Exploring the Internet.* Previously a frequent contributor to the trade press, Malamud has recently devoted his efforts to starting a source of technical news and information about the Internet and networking. Internet Talk Radio distributes audio programs on the Internet about the Internet. He can be reached as: `carl@radio.com`.

Multiprotocol Internets:
Users' Best Hope for Open Systems

by Nick Lippis, Steve Moore, and John P. Morency,
Strategic Networks Consulting, Inc.

Changes

The networking industry is in the midst of significant change on multiple dimensions. During 1993 there will be a whole new tier of mobile computing devices being pushed by Apple, IBM, AT&T, HP and others. These small hand-held computers, commonly referred to as *digital personal communicators,* will be nearly useless without communications, preferably wireless messaging services.

At the same time, a new tier of internetworking products about to be unleashed upon the industry will support branch office or outpost connectivity for corporate enterprise internets. New high end router products from Coral Networks, Cisco Systems and Wellfleet are pushing packet per second rates up into the 200-to-400 thousand range. Switched LAN technologies from companies such as Kalpana are offering faster throughput and a more efficient way to connect multiple LAN segments. "Ethernet on Steroids" is here at both full duplex 20Mbps and now 100Mbps from a wide variety vendors including Cabletron, HP, Grand Junction Networks, etc.

Cell based local area networking hubs from SynOptics, Fore Systems, Hughes LAN Systems, Adaptive, etc., will be available this year offering enhancements to change management not found in today's shared LAN technology. The T1 multiplexer manufacturers are about to offer the industry its next generation intelligent bandwidth managers based on cell switching technology with cell per second rates to the tens of millions. On the wide area side new fractional T3 services are available while frame relay and SMDS are offered by more and more carriers across larger geographies.

Enterprise Networks

But what does all this change mean? Sure, technology marches on, but networking technology hasn't marched this fast before. The bottom line is that our current enterprise networks, which are made up of many of the above technologies, are entering a period of fundamental transition. This transition is one part technology change and one part business re-engineering. A few years ago, the enterprise net used to be thought of as the SNA and Integrated Voice/Data T1 Network. From an organizational point of view, the enterprise network was purchased and managed by central MIS and telecom groups.

Times have changed, to say the least. There are multiple enterprise networks in production use today that have been implemented bottom-up rather than top-down. In many large organizations, there is now the Enterprise Voice Network, the Enterprise Internet and the Enterprise Video Network. Clearly, the term "Enterprise Network" is expanding to include both voice and data desktop communications, as well as new video services and their respective network infrastructure components. The enterprise network includes network interface cards, physical wires, logical networks and management systems, as well as carrier and outsourcer services.

Pricing

One of the major shifts in the traditional Integrated Voice/Data T1 Network was the change in voice messaging pricing. Messaging costs came down along with T1 rates, which motivated most network managers to move their voice networks onto public virtual offerings from the interexchange carriers such as AT&T, MCI and Sprint. Voice is now a commodity, and commodities are not strategic. If you can get a cheaper deal from a carrier, take it.

If voice compression and lower T1 pricing makes it more economical to handle voice through a private network, so be it. However, when staffing costs are factored into the price of a private voice network, 9 times out of 10 it will be best to move voice traffic to the public network.

It is the data-oriented enterprise internet, not the voice network, that creates a competitive advantage when it is architected to leverage the firm's profit drivers. Many companies have used the enterprise internet as an electronic delivery platform for applications that provide both strategic and economic advantages; American Airlines' *Sabre* reservations network was one of the earliest examples of this.

In some cases, a data network service becomes more valuable than the businesses it supports. Telerate, a principal provider of bond pricing data, was acquired in late 1989 at a valuation of over $2 billion, while E.F. Hutton, the second largest brokerage concern in the U.S., was sold at about the same time for $1 billion. Some local exchange carriers derive more profit from their *Yellow Pages* services than from selling dial-tone. Similarly, some publications and transaction services that capture, process and recycle information have become more valuable than the operations the information was originally generated to support. The enterprise internet is strategic because it is the vehicle for all of this high-value information distribution.

Shift in focus

At the same time that users' enterprise internets were evolving into tripartite architectures characterized by largely separate voice, data, and video subnets, centralized computing and applications resources were becoming decentralized. The old hierarchical model of centralized mainframe computers accessed by many dumb terminals gave way to peer-to-peer distributed computing. Businesses in nearly every vertical market have moved computing resources closer to the end user. Once the CPU cycles have been distributed, the next step is to distribute the applications too, so that the mainframe is relegated to the role of a server commanded by increasingly intelligent clients.

The mainframe's decline has enabled Microsoft to sell approximately a million copies of Windows a month into a worldwide installed base of some 50–60 million desktop and laptop computers worldwide, while IBM recently posted the biggest annual loss in corporate history and announced that it would lay off approximately 25,000 employees from its mainframe groups.

Purchase decoupling

As more and more desktop computers were linked to LANs interconnected through internetworking devices such as bridges, routers and internet nodal processors, a decoupling of computer purchasing and network purchasing occurred. The model of purchasing a computer and its associated networking from one vendor gave way to purchasing the right computer for each individual user, while purchasing networking devices and services from third parties. This trend was accelerated by the failure of IBM, DEC and other big systems vendors to deliver on their promises to transform their proprietary computer networking architectures into OSI-compliant environments capable of accommodating a wide range of diverse computing, transmission and switching devices.

Into this breach leapt a host of smaller, more nimble companies that began providing users with more versatile networking products whose protocol, configuration, interconnection and management features rapidly outstripped the large systems vendors in terms of price and performance.

continued on next page

Multiprotocol Internets *(continued)*

The low cost and high performance of LANs enabled users to eliminate their single-vendor networks and build interconnected LANs populated by diverse computing platforms and peripherals.

This "unarchitected" deployment of isolated, non-standards-compliant LANs resulted in a host of problems, including duplication of address numbering for many protocols on the enterprise network. Only now, after five to ten years of incompatible LAN and workgroup computing deployment, are enterprises finally beginning to converge their incompatible systems through network architectures and procurement policies that mandate vendor compliance with open systems networking standards.

LAN Technology	Desktop Adapter Card	Hub Interface
4 Mbs Token Ring	**$650**	**$150**
10BaseT	$150	$60-to-$40*
16 Mbs Token Ring	**$650**	**$150**
Switched Ethernet	$150 (10BT)	$800-to-$400*
Multipoint Routing	$150 (10BT)	$400-to-$200*
100 Mbs Ethernet	$200	$250*
100 Mbs ATM	$500	$1,000*
100 Mbs FDDI (UTP)	$1,000	$1,000

* During 1993

Table 1: LAN Techonoly Cost Comparisons

Many choices

In the 1990s, LANs are evolving so rapidly that many user organizations are hard-pressed to make wise purchase decisions (Table 1). 1993 will bring 100Mbps Ethernet at price points in the $400 range, for both adapter board and hub. Further, Cabletron has been pushing full duplex Ethernet running at 20Mbps, while switched Ethernet technology has become increasingly viable as a means to raise throughput and reduce network delay. FDDI over copper adapters will be priced at under $1k this year. All this adds up to lower cost, higher speed networks.

But one LAN technology is conspicuous by its absence from the above list. While competing technologies have undergone functionality upgrades and price declines, Token Ring technology has not been significantly changed since its introduction, and prices for Token Ring adapters and hub ports are far more expensive than alternative LANs. Yes, there has been an increase in the use of TCP/IP and other open protocols on rings. And Token Ring adapters now appear on workstations from Sun, HP, and IBM. But these workstation vendors view Token Ring mainly as an IBM interconnect technology, not a vendor-independent LAN alternative.

Token Ring will play a decreased role in corporate networks of the future. This is not to say that Token Ring bridges, routers, hubs and the interconnection of Token Ring with other LAN technologies will go away. On the contrary, these technologies will continue to grow in market need. However, the growth of new Token Ring nodes will start to level off.

Other network options will squeeze Token Ring into specific niches. At the low end, 4Mbps Token Ring, intended to link desktops, will be squeezed by faster and cheaper alternatives such as 10BaseT and potentially 100Mbps Ethernet. At the high-end, 16Mbps Token Ring, originally planned for use in backbone networks, will be squeezed by much faster alternatives: FDDI and ATM.

New multiprotocol backbone products

Thanks to the unarchitected LAN proliferation noted above, users will be forced to make their Ethernet, Token Ring and FDDI LANs coexist for the foreseeable future. Each of these key LAN technologies is evolving to support multiple higher-level protocols such as TCP/IP, DECnet, AppleTalk and IPX. Therefore, when connecting LANs over the wide area, a multiprotocol backbone is also needed.

The first popular LAN interconnection technology was the bridge, which preserved the multiprotocol aspect of LANs even when they were extended over the wide-area. Now that most large enterprises are converting from bridges to routers for LAN interconnection, multiprotocol routers have become the rage. The number of protocols supported in routers will continue to increase during 1993, as will the router vendors' sophistication in handling various protocols. Many LAN-based protocols were not designed to operate over a large enterprise network and will require a variety of adaptation mechanisms—such as broadcast suppression and encapsulation—to make them work well.

SNA

The last major protocol to appear on the multiprotocol internet is IBM's *Systems Network Architecture* (SNA). Since corporate internets made up of multiprotocol routers with concurrent bridging are so flexible in their support of a wide range of protocols and networking topologies, it is only a matter of time until currently emerging technical alternatives allow SNA to be assimilated into the corporate internet architecture in most enterprises. The dedicated SNA backbone will be replaced over the next few years by a multiprotocol backbone.

Internetworking Complexity Crisis

As network managers know only too well, networks inevitably get larger and faster over time. Enterprise Internets are no exception to the rule, and in fact they are growing faster than most users can manage and at times near the point of chaos. Consider that 10BaseT is the new computer network interface replacing RS-232C, representing nearly a two order of magnitude increase in bandwidth for every desktop plugged into the enterprise network. But the complexity crisis doesn't come solely from increased bandwidth, it comes more from managing multiple logical networks.

As the number of protocols or logical networks increases, so does the complexity. In a multiprotocol internet, there must be multiple naming and numbering plans, along with enough staff to plan and manage multiple logical networks. Large router network configuration and management is a demanding art with few experienced practitioners. There is a clear lack of network design and management tools for large router networks, some of which today have more than 1,000 routers. With this increase in the use of router technology at a time when there is a lack of management tools, one has to ask whether the Enterprise Internet is heading into a brick wall.

As the Enterprise Internet increases in size and extends across national boundaries, another order of magnitude of complexity is added in just dealing with the logistics of international carriers and the widely disparate organizational cultures of foreign PTTs. As the complexity increases so does the cost since the primary way to deal with internet complexity is to throw people at the problem. And as we all know, people costs constitute the largest slice of the network ownership pie.

One major complexity management problem users face is that most router vendors view reliability the same way they view network management—as an afterthought.

continued on next page

Multiprotocol Internets *(continued)*

As a result, some network planners are starting to question their commitment to the multiprotocol backbone. True there is a range of benefits, as noted above, but these benefits are negated by the lack of inter-protocol congestion management in today's internets. This leaves network planners whose enterprise internets become congested with only one option: investing in over-capacity in terms of router configurations and wide area bandwidth. Most network planners are starting to think twice about placing SNA over the multiprotocol backbone because the introduction of connection-oriented SNA into connectionless multiprotocol internets has caused some SNA sessions to abort.

Hub-centric architecture

Fortunately for users, it appears that new *smart hub* technologies will ease the internet complexity problem by facilitating both the concentration and interconnection of LANs. The smart hub acts as a simplification agent which provides a structured, modular framework for internet components so that management visibility and control extends down to the level of each device attached to a LAN. The smart hub is a solution to internet reliability and troubleshooting problems.

An essential adjunct to the use of smart hubs is a star topology for wiring. When the older LAN cable topology is folded into a star-wired topology, it is now possible to bring each device's connection into the smart hub on its own wire or lobe. This means that if a wire run breaks, it only affects a single user. It also means that each device can be brought into its own port, with its own monitoring and control capabilities. At its most fundamental level, the hub provides a box, technically speaking a *multiport repeater*, that gives each device its own port. The LAN is thus reduced to a single box to which stations are connected via star-wired cabling, which today is usually unshielded twisted pair (UTP) or shielded twisted pair (STP).

The hubbed approach to Ethernet, Token Ring, FDDI and soon ATM LANs combines the star-wired topology of standard telephone wiring with additional electronics in the repeater that perform per-port monitoring and control. To accommodate large numbers of users at a low cost per port, the hub vendors have developed modular hub architectures with various sizes of chassis that each hold a number of multiport repeater cards. The cards share a common backplane bus in the chassis that they use to communicate, creating a potentially large LAN. Once a general, modular hardware structure is in place, it is a natural next step to add cards that perform other LAN functions, such as terminal servers, wide-area network interfaces and management reporting. The leading hub vendors have steadily increased the number of LAN technologies supported, with 10BaseT, Token Ring (802.5), FDDI, and LocalTalk for Apple Macintoshes being the most frequently supported.

Hub integration

It is now clear that the hub has made the wiring closet the point of integration for active internet and LAN components. There is no real reason why it has to stop there, however. As the enterprise internet evolves into an enterprise distributed computing utility, there will be opportunities to incorporate various kinds of servers and gateways into the hub. Although it started out as a solution to the problems of cable-based Ethernets, the hub can be a strategic technology and a key enabler of distributed computing. Figure 1 summarizes the key steps in the expansion of the hub concept.

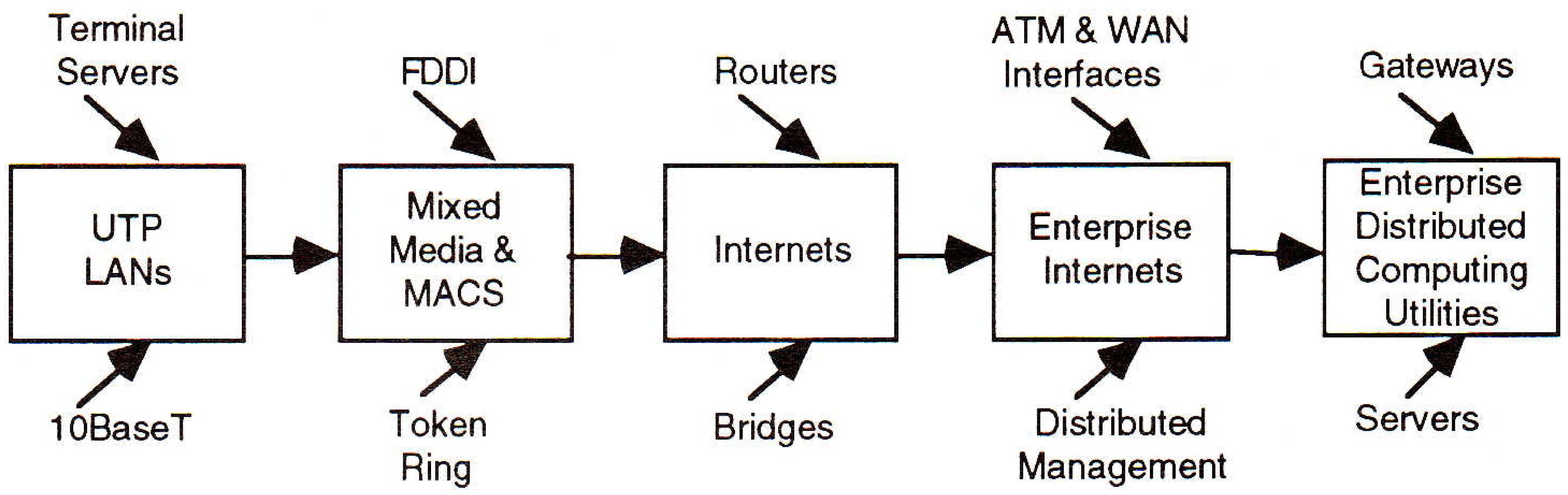

Figure 1: Evolution of the Hub Concept

Currently, hub and router vendors are collaborating to fully integrate routing technology into hubs. This technology integration into the hub does not stop at physical connectivity and logical networks, but is also extended into server functionality. Hub vendors are increasing the amount of computer power, memory and storage technology in these hub systems so that a variety of servers (naming, file, print, gateway, etc.) can be offered at the hub level. Networth of Texas is a company that is pushing the computing hub concept. If stock price is a bellwether to this growing hub trend then consider the following: Networth's stock price during its initial public offering was $15 on the morning of November 23, 1992 and closed near $30 the same day.

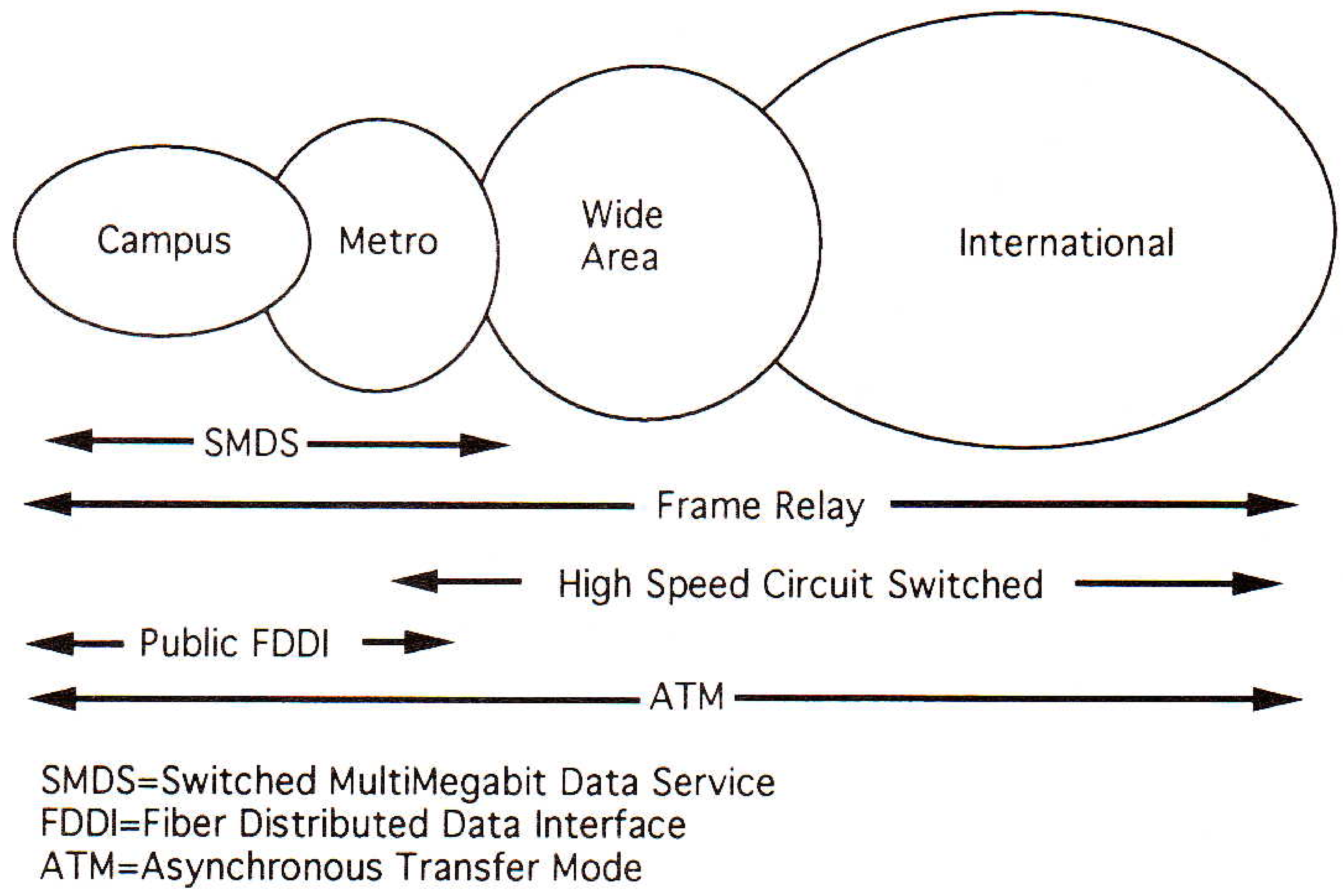

Figure 2: Virtual Data Networks Emerge

Virtual Data Networks

Simultaneously with the above-noted evolution of LANs and multiprotocol internets, there has been an enormous amount of technology change in the wide area, primarily due to the explosive growth in user demand to link LANs on a national or even global scale. After two decades of sitting on the sidelines of the data networking world, the carrier community is aggressively deploying virtual data services on a worldwide scale. These services offer the prospect of lower inter-building communication costs through lower pricing and partial outsourcing of personnel. The key technical benefits of virtual data networks are:

- Full interconnection of all sites on the virtual network, simplifying routing decisions.
- Bandwidth on demand, i.e., flexible use of high bandwidth services rather than dedicated private lines of fixed capacity.
- Better management and control capabilities than older private line services.

continued on next page

Multiprotocol Internets *(continued)*

- Economies of scale from sharing of bandwidth among large numbers of users in many enterprises.
- More direct inter-enterprise networking using the public virtual networks as a common point of interconnection.

Virtual data services are being designed for metropolitan area, wide area and international communications. Figure 2, on the previous page, illustrates the geographic extent of emerging virtual data networks.

Economics

Driven by the increased need for enterprise-wide LAN traffic, the emergence of public virtual data networks and lower-priced voice messaging, a new backbone interconnect device is emerging. Most corporations are disintegrating their T1 multiplexed integrated voice and data networks primarily due to increased LAN traffic flows (30–40% increases per year) and major shifts in tariff economics (lower costs for voice messaging and private digital bandwidth). As a result, there is a convergence of telecommunications technology and hub technology taking place. The hub and routing vendors are starting to provide what the T1 multiplexer vendors have long provided—near fault-tolerant hardware with redundant power supplies, hot swap capability and redundant modules with one by one automatic switch-over. This is an important technology trend that will allow enterprise networks based on the interconnection of LANs to mature into production quality networks with far higher reliability.

While there will be many announcements of virtual data services based on cell relay or ATM technology during 1993 and 1994, don't expect these WAN options to play a large part in enterprise networks any time soon. The Regional Bell Operating Companies are deploying SMDS slowly during '93 and '94, while MCI has promised to make SMDS available during 1994 on a national level. AT&T has made public its software-defined broadband network to be available during 1994 and 1995. Sprint will be offering a public ATM service in 1994.

Before these cell-based virtual data services come to market on a broad scale, there is a dynamic occurring in the private line market. T3 pricing has been increasing in past months while T1 pricing continues to fall. This divergence between T3 and T1 pricing is making T3 services harder to cost justify, thus creating a market for Fractional T3 services. The authors believe that carriers are deliberately creating a situation in which pent-up demand for WAN bandwidth between T1 and T3 will stimulate demand for future cell-based services. By increasing T3 (and fractional T3?) pricing, pent-up demand can be created for cell-based services that will vary in speed from 1.5 Mbs up to 100s of Mbs.

Private wide-area broadband networks will be difficult to cost justify during 1993 and 1994 due to the expense of high speed WAN services. Cost justification will have to be based on metrics such as the cost per cell and bulk volume discounts on T1, creating an enterprise network that lets networked applications perform without geographic bounds.

Broadband networks

Over the next five years, a variety of new public and private networking technologies will be introduced that carry data networking into the 100Mbps range and beyond. The primary technology enabler of broadband networking is *fiber optics,* which offers the promise of giga-to-tera bit per second networking. While fiber optics is the technology enabler of broadband, the business driver for implementing broadband will be the increasing use of visual information.

Video, still image and 3-D CAD will dramatically increase the need for bandwidth, creating a demand for 100Mbps or greater services along certain data networking corridors in the enterprise network. The first widely used fiber-based broadband technology is FDDI, a 100Mbs shared medium LAN technology.

SONET

Other broadband technologies are under development. At the transmission level, the *Synchronous Optical Network* (SONET) standards are nearly defined and endorsed worldwide, but SONET rollout will be slow at best. At the next higher level, new switching technologies are being developed that allow for hardware-based routing of data through a switch. These new *Asynchronous Transfer Mode* (ATM) switches employ small, fixed length data fragments called *cells* rather than longer, variable length frames or packets used in today's bridges and routers.

ATM

ATM is poised to be the switching technology that can unify the tripartite enterprise voice, video and data communication internets discussed above. Vendors such as Newbridge, Timeplex, NET, StrataCom, and others will be introducing new-generation enterprise network switches based on ATM technology during 1993. As key devices for cell based enterprise broadband networks, these intelligent bandwidth managers will interconnect local-area ATM and legacy networks (including TDMs, PBXs, video codecs, etc.) together over the wide area.

ATM will be implemented first in the local area because private wide-area broadband networks will be prohibitively expensive for at least the next few years unless leased-line and VPN tariffs decline more rapidly than expected. Also, it is in the local area where the power of cell based broadband technology can be best applied.

Hub vendors are working to incorporate broadband network switching technology into their hub systems. The core switching fabric for smart hubs in the 1993/4 time frame will most likely be ATM or cell relay switching technology. ATM support in hubs has been announced by a wide range of networking vendors, and it is widely endorsed as the architecture of choice for the public carriers. On the desktop, some believe that ATM interface boards for workstations and next generation personal computers will be competitive in price with FDDI by the mid-1990s.

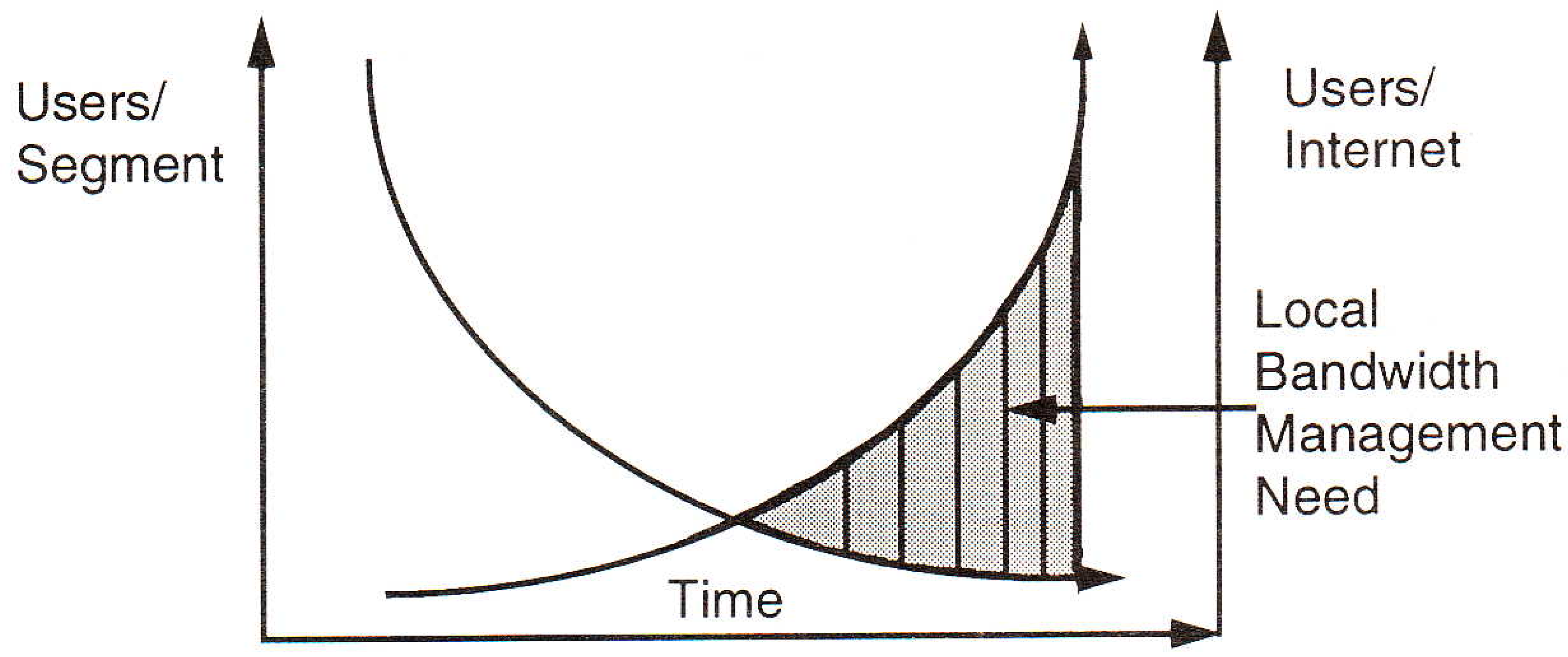

Figure 3: Driver for ATM LANs

Figure 3 illustrates the factors driving the requirement for local ATM networks. The number of users per LAN segment continues to decrease year after year after year. So in 1990 there may have been 20 users per LAN segment; in '91 there may have been 15; in '92 seven; in '93 five; in '94 three; and in '95 one. As this trend continues to play itself out, the LAN ceases being a building traffic distribution technology and becomes an interface.

continued on next page

Multiprotocol Internets *(continued)*

This is driven mainly by the need for uncontended bandwidth at the local level. At the same time there is an increase in the number of user LAN attached devices being plugged into the enterprise internet. This is mainly due to lower interface costs, for example, 10BaseT at $40 per port and under $100 for Ethernet adapters. The divergence of these two curves is creating a need for local bandwidth management. The technology we use today is repeaters, multiport bridges, multiport routers, Ethernet switches and increasingly ATM in hubs. ATM offers an efficient way to provide segment switching between desktop LANs.

ATM logical-to-physical mapping

Even more important than the above physical restructuring taking place within building and campus networks is the interaction of the overriding logical networks with the underlying physical networks. To increase the reliability and integrity of today's building and campus networks so that they can be the infrastructure of tomorrow's enterprise distributed computing utility, an alignment of physical and logical networks needs to occur. This is exactly what ATM LANs—and to a lesser degree, switched Ethernet LANs—offer through the virtual LAN, which is a broadcast group that creates a mapping between various ports on ATM hubs. The ATM hub provides both switched and permanent virtual circuits between ports. By linking multiple clients and servers together through a virtual LAN, physical and logical networks can be aligned again. For example, if NFS, LAN Manager, NetWare, VAX and AppleShare servers were connected to an ATM hub, the hub would provide either permanent or virtual circuits between these servers and their respective clients, thus removing the overlap among multiple logical networks.

Change Management

ATM hubs also promise to revolutionize change management. For example, let's assume that a workstation has an interface to an ATM hub. The ATM hub also has many LAN ports on it, including Ethernet, Token Ring, FDDI and Apple LocalTalk. The ATM hub can establish 10, 16, 50, 100+ Mbps connections between the workstation and various attached LANs forming a virtual LAN. Now let's assume that the workstation user has to move to the second floor of a building. At the ATM hub management console, all the network manager has to do is point and click on the workstation, move the icon to the new location and its entire connectivity map is automatically established in real time. Or he can simply move the workstation to its new location and plug it in to the ATM hub, and it automatically establishes its previous connections. No pulling of wires, changing logical addresses, re-setting network services on servers, etc. Change management becomes easy.

This has even a more significant impact if this workstation is a server to many clients. ATM hubs then may allow server placement to be independent of building and campus location. This also offers a significant advantage to network managers who wish to implement a "centralized" distributed computing utility. That is having the option to centralize servers in a glass house—yes, a glass house—and provide huge dynamic pipes into it from various LAN segments across the building and campus.

It is interesting to ponder the role of routers in an ATM LAN environment. Some observers say that stand-alone routers will be cannibalized as routing technology becomes interbred with hubs, ATM hubs and high end ATM switches. Then routing will take place on the fringes of the enterprise network.

Some observers say that router vendors may even get out of the hardware business and focus instead on higher value software, as did Novell during the 1980s, providing their code on the best communication hardware platform available.

Mobile computing

As noted in the introduction, a new tier of wireless, mobile networking technologies will proliferate during the next few years, driven by a proliferation of ever-smaller computers and *personal digital assistants* (PDAs). These devices also must be integrated into the enterprise network. Different technologies will be used for wireless networking inside buildings versus in the open. Today's cellular telephone services will be augmented by digital cellular and *Personal Communication Network* (PCN) systems during the next few years. Wireless communication technologies will link small remote offices and mobile individuals to the enterprise network.

In addition to wireless mobile computing, there are many other future applications that will drive user companies to build Enterprise Broadband Networks. These applications include distributed databases, e-mail and digital video computing, which promises to deliver both playback and real-time video conferencing services at the desktop.

Location independent objects also offer a major challenge to the enterprise network architect. For example, let's assume that a video based object resides on a U.S. server within a worldwide enterprise network. If users in Tokyo start to interact with this video object through their desktops, users in other countries worldwide could experience serious performance degradation across international links.

Conclusions

The changes delineated above will require network planners to make critical decisions concerning the architectures of their multiprotocol enterprise internets. Network architecture is the bridge between network strategy and evolution. Architecture provides a framework within which networking investment decisions can be made based on a mapping of the firm's overall business strategy to its network strategy. This is a very end user or business unit driven way to develop network architecture. It assures that the enterprise network architecture will be an enabler of the firm's profit drivers.

The single most important advantage of multiprotocol enterprise internets is that they offer users a migration path toward open networks. Supporting both older proprietary protocols and newer open systems protocols on the same internet is the best way for a highly diverse networked computing environment to evolve toward open systems. The multiprotocol internet allows the enterprise management group to shut off proprietary protocols and turn on open protocols as the application environment slowly migrates toward open platforms, and as investments in older systems are written off. When coupled with a strategy of employing dual protocol stacks (one proprietary and one open) in key centralized systems, the multiprotocol internet becomes a dynamic open systems environment that enables the enterprise to manage its growth wisely.

References

[1] Malis, A., "Multiprotocol Encapsulation over Frame Relay," *ConneXions,* Volume 6, No. 8, August 1992.

[2] Kozel, E., "The Cisco/DEC/NTI/StrataCom Frame Relay Specification," *ConneXions,* Volume 5, No. 3, March 1991.

[3] Vair, D., "Components of OSI: X.25—the Network, Data Link, and Physical Layers of the OSI Reference Model," *ConneXions,* Volume 4, No. 12, December 1990.

continued on next page

Multiprotocol Internets *(continued)*

[4] Falk, G., "LAN Interconnection Over X.25," *ConneXions,* Volume 7, No. 2, February 1993.

[5] Wolter, M., "Fiber Distributed Data Interface (FDDI)—A Tutorial," *ConneXions,* Volume 4, No. 10, October 1990.

[6] Hughes, L., & Starliper, S., "Switched Multimegabit Data Service (SMDS), *ConneXions,* Volume 4, No. 10, October 1990.

[7] Lippis, N., & Herman, J., "Widening Your Internet Horizons: Wide-Area Options for Internets," *ConneXions,* Volume 5, No. 10, October 1991.

[8] Krishnaswamy, P., & Ulema, M., "Developments in SMDS," *ConneXions,* Volume 5, No. 10, October 1991.

[9] Potts, J., "Moving Towards the Ultimate Telecommunications: Personal Communications Services (PCS)," *ConneXions*, Volume 6, No. 10, October 1992.

[10] Joyce, Steven T. & Walker II, John Q., "Advanced Peer-to-Peer Networking (APPN): An Overview," *ConneXions,* Volume 6, No. 10, October 1992.

[11] Clark, Wayne, "SNA Internetworking," *ConneXions,* Volume 6, No. 3, March 1992.

[12] Clark, Wayne, "Accommodating SNA Peer-to-Peer Networking in a Multiprotocol Environment," *ConneXions,* Volume 7, No. 3, March 1993.

NICHOLAS J. LIPPIS III is Founder and President of Strategic Networks Consulting, Inc. (SNCI). His areas of expertise are in the technical and business aspects of the design and management of enterprise networks, including advanced Local Area Network (LAN) interconnect architecture and implementation alternatives, wide area voice/data integrated transmission systems, fiber optic networks, tariff and competitive analysis. Mr. Lippis is a contributing editor and monthly columnist for *Data Communications* Magazine. He is also a member of the technical advisory board to the INTEROP conference. Prior to SNCI, Mr. Lippis was a Principal Consultant with Northeast Consulting Resources, Inc., where he developed a highly successful LAN interconnect practice. Mr. Lippis was a Strategic Analyst and Telecommunications Consultant within Digital Equipment Corporation's Networks & Communications Engineering and Information Systems organizations, respectively. Mr. Lippis received the prestigious Digital Information Systems Achievement award for his contribution to the design and installation strategy of Digital's 400 mile private fiber optic network. This project required a capital expenditure of $25 million and will yield a displaced cost savings of $155 million over a ten-year period.

STEVE MOORE is editor and publisher of *The Internetwork Advisor,* a monthly newsletter produced by Strategic Networks Consulting, Inc. (SNCI). Before joining SNCI, Mr. Moore was features editor for *Network World* and editor-in-chief of *Global Networks* magazine, a globally-distributed publication serving corporate executives who manage worldwide communications networks. Prior to that, he was a senior writer for *Computerworld.* Mr. Moore holds a master's degree in journalism and telecommunications from the University of Colorado.

JOHN P. MORENCY is a Principal Consultant with Strategic Networks Consulting, Inc. (SNCI). His expertise includes client-server computing, network management systems, online information access and retrieval, enterprise messaging systems, legacy protocols, LAN internetworking, transaction processing systems, and PC communications. His experience includes nearly 20 years of product development and end user networking, both domestically and internationally. He comes to SNCI from the FW Faxon Company where he successfully transitioned Faxon's computing infrastructure from a centralized mainframe time-sharing environment into a distributed client–server system that supported more than 1,000 online users. Prior to joining Faxon, he served as Technical Director for the Open Network Systems (ONS) Product Creation Unit (PCU) of Digital Equipment Corporation, which included product responsibility for Advantage Networks, DEC–IBM Internetworking, Messaging, TCP/IP, Token Ring Support and Network Management Systems.

Book Review

The *Internet System Handbook,* edited by Daniel Lynch and Marshall Rose, Addison-Wesley, 1992, ISBN 0-201-56741-5.

Organization

The Internet System Handbook (ISH) is a compilation of 19 chapters each written by a different author. The book provides insight and philosophy on the evolution of the Internet, protocols currently in use, and concerns for the future. For the most part I found the individual chapters to be well written. The book is designed to be used as a reference, and each chapter is entirely self-contained. It is organized into four major sections: Introduction, Technologies, Infrastructure, and Directions. I'll give a brief overview of the elements of each section, and follow up with my overall impression.

"Introduction" offers an interesting history of the evolution of the Internet. It also describes the process for adding/incorporating new standards for—and the internationalization of—the Internet.

"Technologies" covers the protocols that are currently in use. This includes the core TCP/IP protocols, and routing protocols. It also has information on the major applications currently in use (SMTP, FTP, Telnet), issues regarding protocol-level security, and a rogue chapter on developing networked applications.

"Infrastructure" covers a number of loosely related topics under the general heading of technology management. A quick hit list is: an overview of directory services (*whois, finger, WAIS*), SNMP, problem tracking in an internet, a nice chapter on IP network performance, and security issues.

"Directions" covers the future of the network, and problems that need to be dealt with in order for the next generation of the Internet to be successful. One chapter is devoted to the problem of running out of IP addresses, and the other to changes in the needs of Internet users.

The ISH is certainly grand in scope and covers a wide number of topics. In reading it I found myself faced with the problem of figuring out who the intended audience is. The book goes from very low-level issues (application development and core protocols) to very high-level issues (problem tracking over an Internet backbone, and privacy issues with regard to directory services).

Some bad

The organization is a bit lacking. Chapters seem to be ordered more by their title than actual content. Some later chapters cover more background on a topic than earlier ones. I also found that about half of the chapters contain what I consider to be "read once" information.

Some good

The book offers a real picture of what the Internet is like, along with issues that are being faced by the Internet community. The chapters present tradeoffs of various problems and solutions. The book is very good for bringing someone completely up to speed on issues in the Internet (or similar networks) and the process of change.

Caveats

This is not a "how-to" book. It describes the function of available services and programs in an informational way. It does give the details of exactly how to use those services. The chapter on network programming seemed out of place with the rest of the book as it is very much a "how-to" chapter.

Overall decent

The ISH would be excellent for someone who needs to know the current state of the Internet. There are many chapters to choose from, and there is probably a large enough subset of chapters to make the book worthwhile to a number of audiences. This book rates about a B.

—*Eli Charne, UC Irvine*

Internet Domain Survey—January 1993

Introduction

The Domain Survey attempts to discover every host on the Internet by doing a complete search of the *Domain Name System.* The latest results gathered during mid-January 1993 are listed. For more information see RFC 1296; for detailed data see the `pub/zone` directory on `ftp.nisc.sri.com`. This survey was done using the census program developed at the University of California at Santa Cruz; see technical report UCSC-CRL-92-34 on host `ftp.cse.ucsc.edu`. The statistics below were generated by running the collected host data through a number of utility programs.

—Mark Lottor
Network Information Systems Center
SRI International

Host and Domain Count

	Jan 93	Oct 92	Jul 92	Apr 92	Jan 92	Change (Jan – Jan)
Hosts:	1,313,000	1,136,000	992,000	890,000	727,000	80.6%
Domains:	21,000	18,100	16,300	20,000	17,000	23.5%

Number of Networks (based on DNS IP addresses)

	Jan 1993	Oct 92	Jul 92	Change (Jul – Jan)
Class A:	54	52	60	–9.0%
Class B:	3206	2985	2714	18.1%
Class C:	4998	4468	3795	31.7%
Total:	8258	7505	6569	25.7%

Host Distribution by Top-Level Domain Name and Percent Change since January 1992

410940	edu	69%	23581	ch	86%	3542	kr	136%	782	is	*	29	cy	*
347486	com	92%	23197	jp	171%	3451	hk	684%	692	us	475%	17	my	*
79772	gov	72%	20109	no	97%	2418	be	588%	610	hu	*	13	tn	–52%
67111	de	116%	16356	fi	36%	2053	nz	84%	349	cl	*	11	yu	*
62327	mil	127%	9986	net	26%	1912	cs	*	121	lu	*	8	lv	*
61429	au	94%	9052	at	172%	1910	br	536%	112	ve	*	5	th	*
58431	uk	208%	7834	it	188%	1882	pt	141%	105	ar	*	5	gb	*
52755	ca	95%	5911	es	256%	1663	pl	*	89	ee	*	4	aq	*
31490	org	64%	5459	dk	204%	1365	sg	182%	79	in	*	3	cr	*
26014	fr	100%	4356	za	368%	1330	ie	259%	63	su	–67%	1	si	*
25991	se	40%	4143	il	104%	1239	mx	339%	58	int	n/a	1	bg	*
25665	nl	101%	4021	tw	398%	860	gr	160%	45	ec	*			

[* = over 1000%]

Top 50 Host Names

633 venus	475 mac2	380 mac4	326 mac11	311 sirius
595 cisco	452 pc2	379 eagle	326 hermes	311 mac9
590 pluto	443 mercury	358 mac5	324 mac7	311 calvin
562 mars	443 iris	358 gauss	323 merlin	302 mac14
527 pc1	435 charon	354 mac10	321 mac12	301 mac15
522 zeus	411 mac3	338 mac6	320 thor	300 athena
519 gw	409 orion	338 hobbes	319 mac8	292 mac16
496 jupiter	385 pc3	335 pc4	319 mac13	289 phoenix
494 mac1	382 newton	332 apollo	318 alpha	285 pc5
484 saturn	381 neptune	330 fred	313 titan	284 gateway

Frequently Asked Questions about the Domain Survey

What do all those domain names stand for?

See `pub/zone/iso-country-codes` on `ftp.nisc.sri.com`.

Why does the domain count go up and down?

I don't know. Do you want to count them?

Are all those hosts really on the Internet?

You would have to *ping* them to find out. If they each took 100 milliseconds to reply to a *ping*, you could find out in only 37 hours.

How many users are on the Internet?

Some people estimate around 10 per host (13 million people). If all of them were appropriately registered, the birthday-daemon would have to deliver 35,616 e-mail messages every day.

Where can I get more information?

See the `pub/zone` directory on `ftp.nisc.sri.com`.

References

[1] Lottor, M., "Internet Growth (1981–1991)," RFC 1296, January, 1992.

[2] Solensky, F., "The Growing Internet," *ConneXions*, Volume 6, No. 5, May 1992.

[3] Marine, A., "How Did We Get 727,000 hosts?" *ConneXions*, Volume 6, No. 5, May 1992.

[4] Crocker, D. "The ROAD to a New IP," *ConneXions*, Volume 6, No. 11, November 1992.

[5] Mockapetris, P., "Introducing Domains," *ConneXions*, Volume 1, No. 6, October 1987.

[6] Mockapetris, P., "Domain Names: Concepts and Facilities," RFC 1034, 1987.

Announcement and Call for Papers

NSC '93, The Network Services Conference 1993 will be held in Warsaw, Poland, 12–14 October 1993. NSC '93 is being organized by EARN in conjunction with EUnet/EurOpen, NORDUnet, RARE, and RIPE.

Overview

Networking in the academic and research environment has evolved into an important tool for researchers in all disciplines. High quality network services and tools are essential parts of the research infrastructure. Building on the success of the first Network Services Conference in Pisa Italy, NSC '93 will focus on the issue of providing services to customers, with special attention paid to the exciting developments in new global high-level tools. We will address the impact of the new global tools on service development and support, the changing function of traditional tools and services (e.g., archives), new services (e.g., multi-media communications), the future role of the library and the effects of commercialization of networks and network services. Customer support at the institutional and campus level, and the role of support in accessing global services, will also be covered.

Format

Talks, tutorials, demonstrations and other conference activities will address the needs of the research, academic, educational, governmental, industrial, and commercial network communities. The official language of the conference will be English.

Venue

Warsaw, the capital of Poland, lies in the center of the country on the Mazovian Lowland. Located on the banks of the Vistula River, it has a population of 1,700,000. Legends speak of Warsaw's past. One of them tells of a mermaid swimming in the waves of the blue Vistula before Mazovian fishermen and foretelling of the founding of an indestructible city. Another account speaks of the founders of the city, Wars and Sawa, lovers whose names were combined to give the city its name. Today, Warsaw is an important administrative, scientific, cultural and communication center. The city, destroyed during World War II, has been faithfully rebuilt and almost all historic buildings have been restored. The conference will be held at the Victoria Intercontinental Hotel, situated in the heart of the city's business and professional center, within walking distance of Warsaw University and just a minute away from the Opera House, Royal Castle and the Old Town.

Topics

The Program Committee for NSC '93 is soliciting proposals for papers, tutorials and demonstrations in all fields related to network services. Subject areas include, but are not limited to, the following:

- Network Resource Tools
- Network Directory Services
- Multimedia Communications
- Electronic Publishing
- Libraries and Networking
- Special Interest Communities
- Groupware, Cooperative Work over the Network
- Networking for Schools
- User Support
- Delivering Services to the Desktop
- Commercialization of Network Services
- Networking in Eastern and Central Europe

Important dates

Deadline for papers:	8 May 1993
Notification of acceptance:	8 June 1993
Deadline for demonstrations:	3 August 1993
Notification of acceptance (demos):	17 August 1993

Tutorials

There will be tutorial sessions on specific network services as part of the regular conference program. A room will be available for workstations and PCs to be used for demonstrations throughout the conference.

Posters

A poster wall will be available to participants for the display of their posters and projects. Terminals with connectivity to EARN and the Internet will be available to delegates.

More information

Further information will be available through the conference mailing list, `NSC93-L@FRORS12.BITNET`. If you want to make sure you receive registration information as well as the preliminary program and other information of interest to conference participants, join the list by sending e-mail to:

```
LISTSERV@FRORS12.BITNET
```

with the line:

```
SUB NSC93-L Your Name
```

If you have any questions or require any assistance, you can contact the conference organizers at:

NSC '93
EARN Office
c/o CIRCE
BP 167
F-91403 Orsay
FRANCE
Tel.: +33 1 6982 3973
Fax: +33 1 6928 5273
E-mail: NSC93@FRORS12.BITNET
or NSC93@FRORS12.CIRCE.FR

Papers should also be submitted to the above address.

Program Committee

Hans Deckers, France (Chair); Daniel Jozef Bem, Poland; Howard Bilofsky, Germany; Klaus Birkenbihl, Germany; Rob Blokzijl, The Netherlands; Daniele Bovio, France; Paul Bryant, United Kingdom; Vasco Freitas, Portugal; Tomasz Hofmokl, Poland; Dennis Jennings, Ireland; Glenn Kowack, The Netherlands; David Sitman, Israel; Marco Sommani, Italy; Iain Stinson, United Kingdom.

Organizing Committee

Frode Greisen, Denmark (Chair); Paul Bryant, United Kingdom; Hans Deckers, France; Nadine Grange, France; Tomasz Hofmokl, Poland; Tadeusz Wegrzynowski, Poland.

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

ADDRESS CORRECTION
REQUESTED